



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : H04N	A2	(11) International Publication Number: WO 00/28725 (43) International Publication Date: 18 May 2000 (18.05.00)
(21) International Application Number: PCT/US99/26126 (22) International Filing Date: 5 November 1999 (05.11.99) (30) Priority Data: 60/107,463 6 November 1998 (06.11.98) US 60/118,020 1 February 1999 (01.02.99) US 60/118,027 1 February 1999 (01.02.99) US (71) Applicants (for all designated States except US): THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK [US/US]; Broadway and 116th Street, New York, NY 10027 (US). AT & T [US/US]; AT & T Labs, Redbank, NJ (US). IBM [US/US]; T.J. Watson Research Center, Hawthorne, NY (US). (71) Applicant (for US only): EASTMAN KODAK [US/US]; Rochester, NY (US). (72) Inventors; and (75) Inventors/Applicants (for US only): PAEK, Seungyup [KR/US]; Apartment 6J, 530 Riverside Drive, New York, NY 10027 (US). BENITEZ, Ana [ES/US]; Apartment 9F, 400 West 119th Street, New York, NY 10027 (US). CHANG, Shih-Fu [-/US]; Apartment 18K, 560 Riverside Drive, New York, NY 10027 (US). ELEFTERIADIS,		Alexandros [-/US]; Columbia University, New York, NY 10027 (US). PURI, Atul [-/US]; AT & T Labs, Redbank, NJ (US). HUANG, Qian [-/US]; AT & T Labs, Redbank, NJ (US). LI, Chung-Sheng [-/US]; IBM, T.J. Watson Research Center, Hawthorne, NY (US). JUDICE, Charlie [-/US]; Eastman Kodak, Rochester, NY (US). (74) Agent: TANG, Henry; Baker & Botts, LLP, 30 Rockefeller Plaza, New York, NY 10112-0228 (US). (81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published <i>Without international search report and to be republished upon receipt of that report.</i>
(54) Title: VIDEO DESCRIPTION SYSTEM AND METHOD (57) Abstract Systems and methods for describing video content establish video description records which include an object set, an object hierarchy and entity relation graphs. Video objects can include global objects, segment objects and local objects. The video objects are further defined by a number of features organized in classes, which in turn are further defined by a number of feature descriptors. The relationships between and among the objects in the object set are defined by the object hierarchy and entity relation graphs. The video description records provide a standard vehicle for describing the content and context of video information for subsequent access and processing by computer applications such as search engines, filters, and archive systems.		

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

VIDEO DESCRIPTION SYSTEM AND METHOD

SPECIFICATION

CROSS REFERENCE TO RELATED APPLICATIONS

5 This application claims priority to United States provisional patent application Serial No. 60/118,020, filed February 1, 1999, United States provisional patent application serial no. 60/118,027, filed February 1, 1999 and United States provisional patent application serial no. 60/107,463, filed November 6, 1998.

FIELD OF THE INVENTION

10 The present invention relates to techniques for describing multimedia information, and more specifically, to techniques which describe video information and the content of such information.

BACKGROUND OF THE INVENTION

With the maturation of the global Internet and the widespread employment of regional networks and local networks, digital multimedia information has become increasingly accessible to consumers and businesses. Accordingly, it has become progressively more important to develop systems that process, filter, search and organize digital multimedia information, so that useful information can be culled from this growing mass of raw information.

20 At the time of filing the instant application, solutions exist that allow consumers and business to search for textual information. Indeed, numerous text-based search engines, such as those provided by yahoo.com, goto.com, excite.com and others are available on the World Wide Web, and are among the most visited Web sites, indicating the significant of the demand for such information retrieval technology.

25 Unfortunately, the same is not true for multimedia content, as no generally recognized description of this material exists. In this regard, there have

been past attempts to provide multimedia databases which permit users to search for pictures using characteristics such as color, texture and shape information of video objects embedded in the picture. However, at the closing of the 20th Century, it is not yet possible to perform a general search the Internet or most regional or local
5 networks for multimedia content, as no broadly recognized description of this material exists. Moreover, the need to search for multimedia content is not limited to databases, but extends to other applications, such as digital broadcast television and multimedia telephony.

One industry wide attempt to develop a standard multimedia
10 description framework has been through the Motion Pictures Expert Group's ("MPEG") MPEG-7 standardization effort. Launched in October 1996, MPEG-7 aims to standardize content descriptions of multimedia data in order to facilitate content-focused applications like multimedia searching, filtering, browsing and summarization. A more complete description of the objectives of the MPEG-7
15 standard are contained in the International Organisation for Standardisation document ISO/IEC JTC1/SC29/WG11 N2460 (Oct. 1998), the content of which is incorporated by reference herein.

The MPEG-7 standard has the objective of specifying a standard set of descriptors as well as structures (referred to as "description schemes") for the
20 descriptors and their relationships to describe various types of multimedia information. MPEG-7 also proposes to standardize ways to define other descriptors as well as "description schemes" for the descriptors and their relationships. This description, i.e. the combination of descriptors and description schemes, shall be associated with the content itself, to allow fast and efficient searching and filtering for
25 material of a user's interest. MPEG-7 also proposes to standardize a language to specify description schemes, i.e. a Description Definition Language ("DDL"), and the schemes for binary encoding the descriptions of multimedia content.

At the time of filing the instant application, MPEG is soliciting proposals for techniques which will optimally implement the necessary description
30 schemes for future integration into the MPEG-7 standard. In order to provide such optimized description schemes, three different multimedia-application arrangements

can be considered. These are the distributed processing scenario, the content-exchange scenario, and the format which permits the personalized viewing of multimedia content.

Regarding distributed processing, a description scheme must provide
5 the ability to interchange descriptions of multimedia material independently of any platform, any vendor, and any application, which will enable the distributed processing of multimedia content. The standardization of interoperable content descriptions will mean that data from a variety of sources can be plugged into a variety of distributed applications, such as multimedia processors, editors, retrieval
10 systems, filtering agents, etc. Some of these applications may be provided by third parties, generating a sub-industry of providers of multimedia tools that can work with the standardized descriptions of the multimedia data.

A user should be permitted to access various content providers' web sites to download content and associated indexing data, obtained by some low-level or
15 high-level processing, and proceed to access several tool providers' web sites to download tools (e.g. Java applets) to manipulate the heterogeneous data descriptions in particular ways, according to the user's personal interests. An example of such a multimedia tool will be a video editor. A MPEG-7 compliant video editor will be able to manipulate and process video content from a variety of sources if the description
20 associated with each video is MPEG-7 compliant. Each video may come with varying degrees of description detail, such as camera motion, scene cuts, annotations, and object segmentations.

A second scenario that will greatly benefit from an interoperable content-description standard is the exchange of multimedia content among
25 heterogeneous multimedia databases. MPEG-7 aims to provide the means to express, exchange, translate, and reuse existing descriptions of multimedia material.

Currently, TV broadcasters, Radio broadcasters, and other content providers manage and store an enormous amount of multimedia material. This material is currently described manually using textual information and proprietary
30 databases. Without an interoperable content description, content users need to invest manpower to translate manually the descriptions used by each broadcaster into their

own proprietary scheme. Interchange of multimedia content descriptions would be possible if all the content providers embraced the same content description schemes. This is one of the objectives of MPEG-7.

5 Finally, multimedia players and viewers that employ the description schemes must provide the users with innovative capabilities such as multiple views of the data configured by the user. The user should be able to change the display's configuration without requiring the data to be downloaded again in a different format from the content broadcaster.

10 The foregoing examples only hint at the possible uses for richly structured data delivered in a standardized way based on MPEG-7. Unfortunately, no prior art techniques available at present are able to generically satisfy the distributed processing, content-exchange, or personalized viewing scenarios. Specifically, the prior art fails to provide a technique for capturing content embedded in multimedia information based on either generic characteristics or semantic relationships, or to
15 provide a technique for organizing such content. Accordingly, there exists a need in the art for efficient content description schemes for generic multimedia information.

SUMMARY OF THE INVENTION

20 It is an object of the present invention to provide a description scheme for video content.

It is a further object of the present invention to provide a description scheme for video content which is extensible.

It is another object of the present invention to provide a description scheme for video content which is scalable.

25 It is yet another object of the present invention to provide a description scheme for video content which satisfies the requirements of proposed media standards, such as MPEG-7.

It is an object of the present invention to provide systems and methods for describing video content.

30 It is a further object of the present invention to provide systems and methods for describing video content which are extensible.

It is another object of the present invention to provide systems and method for describing video content which are scalable.

It is yet another object of the present invention to provide systems and methods for describing video content which satisfies the requirements of proposed media standards, such as MPEG-7

In accordance with the present invention, a first method of describing video content in a computer database record includes the steps of establishing a plurality of objects in the video; characterizing the objects with a plurality of features of the objects; and relating the objects in a hierarchy in accordance with the features. The method can also include the further the step of relating the objects in accordance with at least one entity relation graph.

Preferably, the objects can take the form of local objects (such as a group of pixels within a frame), segment objects (which represent one or more frames of a video clip) and global objects. The objects can be extracted from the video content automatically, semi-automatically, or manually.

The features used to define the video objects can include visual features, semantic features, media features, and temporal features. A further step in the method can include assigning feature descriptors to further define the features.

In accordance with another embodiment of the invention, computer readable media is programmed with at least one video description record describing video content. The video description record, which is preferably formed in accordance with the methods described above, generally includes a plurality of objects in the video; a plurality of features characterizing said objects; and a hierarchy relating at least a portion of the video objects in accordance with said features.

Preferably, the description record for a video clip further includes at least one entity relation graph. It is also preferred that the features include at least one of visual features, semantic features, media features, and temporal features. Generally, the features in the description record can be further defined with at least on feature descriptor.

A system for describing video content and generating a video description record in accordance with the present invention includes a processor, a video input

interface operably coupled to the processor for receiving the video content, a video display operatively coupled to the processor; and a computer accessible data storage system operatively coupled to the processor. The processor is programmed to generate a video description record of the video content for storage in the computer accessible data storage system by performing video object extraction processing, entity relation graph processing, and object hierarchy processing of the video content.

In this exemplary system, video object extraction processing can include video object extraction processing operations and video object feature extraction processing operations.

10

BRIEF DESCRIPTION OF THE DRAWING

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

15

Figure 1A is an exemplary image for the image description system of the present invention.

Figure 1B is an exemplary object hierarchy for the image description system of the present invention.

Figure 1C is an exemplary entity relation graph for the image description system of the present invention.

20

Figure 2 is an exemplary block diagram of the image description system of the present invention.

Figure 3A is an exemplary object hierarchy for the image description system of the present invention.

25

Figure 3B is another exemplary object hierarchy for the image description system of the present invention.

Figure 4A is a representation of an exemplary image for the image description system of the present invention.

Figure 4B is an exemplary clustering hierarchy for the image description system of the present invention.

Figure 5 is an exemplary block diagram of the image description system of the present invention.

Figure 6 is an exemplary process flow diagram for the image description system of the present invention.

5 Figure 7 is an exemplary block diagram of the image description system of the present invention.

Figure 8 is an another exemplary block diagram of the image description system of the present invention.

10 Figure 9 is a schematic diagram of a video description scheme (DS), in accordance with the present invention.

Figure 10 is a pictorial diagram of an exemplary video clip, with a plurality of objects defined therein.

15 Figure 11 is a graphical representation of an exemplary semantic hierarchy illustrating exemplary relationships among objects in the video clip of Figure 10.

Figure 12 is a graphical representation of an entity relation graph illustrating exemplary relationships among objects in the video clip of Figure 10.

Figure 13 is a block diagram of a system for creating video content descriptions in accordance with the present invention.

20 Figure 14 is a flow diagram illustrating the processing operations involved in creating video content description records in accordance with the present invention.

25 Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and
30 spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention constitutes a description scheme (DS) for images, wherein simple but powerful structures representing generic image data are utilized. Although the description scheme of the present invention can be used with any type of standard which describes image content, a preferred embodiment of the invention is used with the MPEG-7 standard. Although any Description Definition Language (DDL) may be used to implement the DS of the present invention, a preferred embodiment utilizes the eXtensible Markup Language (XML), which is a streamlined subset of SGML (Standard Generalized Markup Language, ISO 8879) developed specifically for World Wide Web applications. SGML allows documents to be self-describing, in the sense that they describe their own grammar by specifying the tag set used in the document and the structural relationships that those tags represent. XML retains the key SGML advantages in a language that is designed to be vastly easier to learn, use, and implement than full SGML. A complete description of XML can be found at the World Wide Web Consortium's web page on XML, at <http://www.w3.org/XML/>, the contents of which is incorporated by reference herein.

The primary components of a characterization of an image using the description scheme of the present invention are objects, feature classifications, object hierarchies, entity-relation graphs, multiple levels of abstraction, code downloading, and modality transcoding, all of which will be described in additional detail below. In the description scheme of the present invention, an image document is represented by a set of objects and relationships among objects. Each object may have one or more associated features, which are generally grouped into the following categories: media features, visual features, and semantic features. Each feature can include descriptors that can facilitate code downloading by pointing to external extraction and similarity matching code. Relationships among objects can be described by object hierarchies and entity-relation graphs. Object hierarchies can also include the concept of multiple levels of abstraction. Modality transcoding allows user terminals having different capabilities (such as palmpilots, cellular telephones, or different types of personal

computers (PC's), for example) to receive the same image content in different resolutions and/or different modalities.

As described above, a preferred embodiment of the image description system of the present invention is used with the MPEG-7 standard. In accord with this standard, this preferred embodiment uses objects as the fundamental entity in describing various levels of image content, which can be defined along different dimensions. For example, objects can be used to describe image regions or groups of image regions. High-level objects can in turn be used to describe groups of primitive objects based on semantics or visual features. In addition, different types of features can be used in connection with different levels of objects. For instance, visual features can be applied to objects corresponding to physical components in the image content, whereas semantic features can be applied to any level of object.

In addition, the image description system of the present invention provides flexibility, extensibility, scalability and convenience of use. In the interest of enhanced flexibility, the present invention allows portions of the image description system to be instantiated, uses efficient categorization of features and clustering of objects by way of an clustering hierarchy, and also supports efficient linking, embedding and downloading of external feature descriptors and execution code. The present invention also provides extensibility by permitting elements defined in the description scheme to be used to derive new elements for different domains. Scalability is provided by the present invention's capability to define multiple abstraction levels based on any arbitrary set of criteria using object hierarchies. These criteria can be specified in terms of visual features (size and color, for example), semantic relevance (relevance to user interest profile, for example) and/or service quality (media features, for example). The present invention is convenient to use because it specifies a minimal set of components: namely, objects, feature classes, object hierarchies, and entity-relation graphs. Additional objects and features can be added in a modular and flexible way. In addition, different types of object hierarchies and entity-relation graphs can each be defined in a similar fashion.

Under the image description system of the present invention, an image is represented as a set of image objects, which are related to one another by object

hierarchies and entity-relation graphs. These objects can have multiple features which can be linked to external extraction and similarity matching code. These features are categorized into media, visual, and semantic features, for example. Image objects can be organized in multiple different object hierarchies. Non-hierarchical relationships among two or more objects can be described using one or more different entity-relation graphs. For objects contained in large images, multiple levels of abstraction in clustering and viewing such objects can be implemented using object hierarchies. These multiple levels of abstraction in clustering and viewing such images can be based on media, visual, and/or semantic features, for example. One example of a media feature includes modality transcoding, which permits users having different terminal specifications to access the same image content in satisfactory modalities and resolutions.

The characteristics and operation of the image description system of the present invention will now be presented in additional detail. Figs. 1A, 1B and 1C depict an exemplary description of an exemplary image in accordance with the image description system of the present invention. Fig. 1A depicts an exemplary set of image objects and exemplary corresponding object features for those objects. More specifically, Fig. 1A depicts image object 1 (i.e., O1 2 ("Person A"), O2 6 ("Person B") and O3 4 ("People") contained in O0 8 (i.e., the overall exemplary photograph), as well as exemplary features 10 for the exemplary photograph depicted. Fig. 1B depicts an exemplary spatial object hierarchy for the image objects depicted in Fig. 1A, wherein O0 8 (the overall photograph) is shown to contain O1 2 ("Person A") and O2 6 ("Person B"). Fig. 1C depicts an exemplary entity-relation (E-R) graph for the image objects depicted in Fig. 1A, wherein O1 2 ("Person A") is characterized as being located to the left of, and shaking hands with, O2 6 ("Person B").

Fig. 2 depicts an exemplary graphical representation of the image description system of the present invention, utilizing the conventional Unified Modeling Language (UML) format and notation. Specifically, the diamond-shaped symbols depicted in Fig. 2 represent the composition relationship. The range associated with each element represents the frequency in that composition

relationship. Specifically, the nomenclature "0...*" denotes "greater than or equal to 0;" the nomenclature "1...*" denotes "greater than or equal to 1."

In the following discussion, the text appearing between the characters "<" and ">" denotes the characterization of the referenced elements in the XML

5 preferred embodiments which appear below. In the image description system of the present invention as depicted in Fig. 2, an image element 22 (<image>), which represents an image description, includes an image object set element 24 (<image_object_set>), and may also include one or more object hierarchy elements 26 (<object_hierarchy>) and one or more entity-relation graphs 28 (<entity_relation_graph>). Each image object set element 24 includes one or more
10 image object elements 30. Each image object element 30 may include one or more features, such as media feature elements 36, visual feature elements 38 and/or semantic feature elements 40. Each object hierarchy element 26 contains an object node element 32, each of which may in turn contain one or more additional object
15 node elements 32. Each entity-relation graph 28 contains one or more entity relation elements 34. Each entity relation element 34 in turn contains a relation element 44, and may also contain one or more entity node elements 42.

An object hierarchy element 26 is a special case of an entity-relation graph 28, wherein the entities are related by containment relationships. The preferred
20 embodiment of the image description system of the present invention includes object hierarchy elements 26 in addition to entity relationship graphs 28, because an object hierarchy element 26 is a more efficient structure for retrieval than is an entity relationship graph 28. In addition, an object hierarchy element 26 is the most natural way of defining composite objects, and MPEG-4 objects are constructed using
25 hierarchical structures.

To maximize flexibility and generality, the image description system of the present invention separates the definition of the objects from the structures that describe relationships among the objects. Thus, the same object may appear in
different object hierarchies 26 and entity-relation graphs 28. This avoids the
30 undesirable duplication of features for objects that appear in more than one object hierarchy 26 and/or entity-relation graph 28. In addition, an object can be defined

without the need for it to be included in any relational structure, such as an object hierarchy 26 or entity-relation graph 28, so that the extraction of objects and relations among objects can be performed at different stages, thereby permitting distributed processing of the image content.

5 Referring to Figs. 1A, 1B, 1C and Fig. 2, an image object 30 refers to one or more arbitrary regions of an image, and therefore can be either continuous or discontinuous in space. In Figs. 1A, 1B and 1C, O1 2 ("Person A"), O2 6 ("Person B"), and O0 8 (i.e., the photograph) are objects with only one associated continuous region. On the other hand, O3 4 ("People") is an example of an object composed of
10 multiple regions separated from one another in space. A global object contains features that are common to an entire image, whereas a local object contains only features of a particular section of that image. Thus, in Figs. 1A, 1B and 1C, O0 8 is a global object representing the entire image depicted, whereas O1 2, O2 4 and O3 4 are each local objects representing a person or persons contained within the overall
15 image.

Various types of objects which can be used in connection with the present invention include visual objects, which are objects defined by visual features such as color or texture; media objects; semantic objects; and objects defined by a combination of semantic, visual, and media features. Thus, an object's type is
20 determined by the features used to describe that object. As a result, new types of objects can be added as necessary. In addition, different types of objects may be derived from these generic objects by utilizing inheritance relationships, which are supported by the MPEG-7 standard.

As depicted in Fig. 2, the set of all image object elements 30
25 (<image_object>) described in an image is contained within the image object set element 24 (<image_object_set>). Each image object element 30 can have a unique identifier within an image description. The identifier and the object type (e.g., local or global) are expressed as attributes of the object element ID and type, respectively. An exemplary implementation of an exemplary set of objects to describe the image
30 depicted in Figs. 1A, 1B and 1C is shown below listed in XML. In all XML listings

shown below, the text appearing between the characters "<!--" and "-->" denotes comments to the XML code:

```

5      <image_object_set>
        <image_object id="O0" type="GLOBAL"> </image_object> <!-- Photograph -->
        <image_object id="O1" type="LOCAL"> </image_object> <!-- Person A -->
        <image_object id="O2" type="LOCAL"> </image_object> <!-- Person B -->
        <image_object id="O3" type="LOCAL"> </image_object> <!-- People -->
      </image_object_set>

```

As depicted in Fig. 2, image objects 30 may for example contain three
 10 feature class elements that group features together according to the information
 conveyed by those features. Examples of such feature class elements include media
 features 36 (<img_obj_media_features>), visual features 38
 (<img_obj_visual_features>), and semantic features 40 (<img_obj_media_features>).
 Table 1 below denotes an exemplary list of features for each of these feature classes.

15 **Table 1: Exemplary Feature Classes and Features.**

Feature Class	Features
Semantic	Text Annotation, Who, What Object, What Action, Why, When, Where
Visual	Color, Texture, Position, Size, Shape, Orientation
20 Media	File Format, File Size, Color Representation, Resolution, Data File Location, Modality Transcoding, Author, Date of Creation

Each feature element contained in the feature classes in an image
 object element 30 will include descriptors in accordance with the MPEG-7 standard.
 Table 2 below denotes exemplary descriptors that may be associated with certain of

the exemplary visual features denoted in Table 1. Specific descriptors such as those denoted in Table 2 may also contain links to external extraction and similarity matching code. Although Tables 1 and 2 denote exemplary features and descriptors, the image description system of the present invention may include, in an extensible and modular fashion, any number of features and descriptors for each object.

Table 2: Exemplary Visual Features and Associated Descriptors.

Feature	Descriptors
Color	Color Histogram, Dominant Color, Color Coherence Vector, Visual Sprite Color
10 Texture	Tamura, MSAR, Edge Direction Histogram, DCT Coefficient Energies, Visual Sprite Texture
Shape	Bounding Box, Binary Mask, Chroma Key, Polygon Shape, Fourier Shape, Boundary, Size, Symmetry, Orientation

The XML example shown below denotes an example of how features and descriptors can be defined to be included in an image object 30. In particular, the below example defines the exemplary features 10 associated with the global object O0 depicted in Figs. 1A, 1B and 1C, namely, two semantic features ("where" and "when"), one media feature ("file format"), and one visual feature ("color" with a "color histogram" descriptor). An object can be described by different concepts (<concept>) in each of the semantic categories as shown in the example below.

```

<image_object id="O0" type="GLOBAL"> <!-- Global object: Photograph -->
  <img_obj_semantic_features>
    <where>
      <concept> Columbia University, NYC </concept>

```

15

```

                    <concept> Outdoors </concept>
                </where>
                <when> <concept> 5/31/99 </concept> </when>
    5      </img_obj_semantic_features>
          <img_obj_media_features>
                <file_format> JPG </file_format>
          </img_obj_media_features>
          <img_obj_visual_features>
    10      <color>
                <color_histogram>
                    <value format="float[166]"> .3 .03 .45 ... </value>
                </color_histogram>
            </color>
          </img_obj_visual_features>
    15 </image_global_object>

```

As depicted in Fig. 2, in the image description system of the present invention the object hierarchy element 26 can be used to organize the image objects 30 in the image object set 24, based on different criteria such as media features 36, visual features 38, semantic features 40, or any combinations thereof. Each object hierarchy element 26 constitutes a tree of object nodes 32 which reference image object elements 30 in the image object set 24 via link 33.

An object hierarchy 26 involves a containment relation from one or more child nodes to a parent node. This containment relation may be of numerous different types, depending on the particular object features being utilized, such as media features 36, visual features 38 and/or semantic features 40, for example. For example, the spatial object hierarchy depicted in Fig. 1B describes a visual containment, because it is created in connection with a visual feature, namely spatial position. Figs. 3A and 3B depict two additional exemplary object hierarchies. Specifically, Fig. 3A depicts an exemplary hierarchy for the image objects depicted in Fig. 1A, based on the "who" semantic feature as denoted in Table 1. Thus, in Fig. 3A, O3 4 ("People") is shown to contain O1 2 ("Person A") and O2 6 ("Person B"). Fig. 3B depicts an exemplary hierarchy based on exemplary color and shape visual features such as those denoted in Table 1. In Fig. 3B, O7 46 could for example be

defined to be the corresponding region of an object satisfying certain specified color and shape constraints. Thus, Fig. 3B depicts O7 46 ("Skin Tone & Shape") as containing O4 48 ("Face Region 1") and O6 50 ("Face Region 2"). Object hierarchies 26 combining different features can also be constructed to satisfy the requirements of a broad range of application systems.

As further depicted in Fig. 2, each object hierarchy element 26 (<object_hierarchy>) contains a tree of object nodes (ONs) 32. The object hierarchies also may include optional string attribute types. If such string attribute types are present, a thesaurus can provide the values of these string attribute types so that applications can determine the types of hierarchies which exist. Every object node 32 (<object_node>) references an image object 30 in the image object set 24 via link 33. Image objects 30 also can reference back to the object nodes 32 referencing them via link 33. This bi-directional linking mechanism permits efficient transversal from image objects 30 in the image object set 24 to the corresponding object nodes 32 in the object hierarchy 26, and vice versa. Each object node 32 references an image object 30 through an attribute (object_ref) by using a unique identifier of the image object. Each object node 32 may also contain a unique identifier in the form of an attribute. These unique identifiers for the object nodes 32 enable the objects 30 to reference back to the object nodes which reference them using another attribute (object_node_ref). An exemplary XML implementation of the exemplary spatial object hierarchy depicted in Fig. 1B is expressed below.

```
<object_hierarchy type="SPATIAL"> <!-- Object hierarchy: spatial hierarchy -->
  <object_node id="ON0" object_ref="O0"> <!-- Photograph -->
    <object_node id="ON1" object_ref="O1"> </object_node> <!-- Person A -->
    <object_node id="ON2" object_ref="O2"> </object_node> <!-- Person B -->
  </object_node>
</object_hierarchy>
```

Object hierarchies 26 can also be used to build clustering hierarchies and to generate multiple levels of abstraction. In describing relatively large images, such as satellite photograph images for example, a problem normally arises in

describing and retrieving, in an efficient and scalable manner, the many objects normally contained in such images. Clustering hierarchies can be used in connection with the image description system of the present invention to provide a solution to this problem.

5 Figs. 4A and 4B depict an exemplary use of an clustering hierarchy scheme wherein objects are clustered hierarchically based on their respective size (<size>). In particular, Fig. 4A depicts a representation of a relatively large image, such as a satellite photograph image for example, wherein objects O11 52, O12 54, O13 56, O14 58 and O15 60 represent image objects of varying size, such as lakes on the earth's surface for example, contained in the large image. Fig. 4B represents an
10 exemplary size-based clustering hierarchy for the objects depicted in Fig. 4A, wherein objects O11 52, O12 54, O13 56, O14 58 and O15 60 represent the objects depicted in Fig. 4A, and wherein additional objects O16 62, O17 64 and O18 56 represent objects which specify the size-based criteria for the cluster hierarchy depicted in Fig. 4B. In
15 particular, objects O16 62, O17 64 and O18 56 may for example represent intermediate nodes 32 of an object hierarchy 26, which intermediate nodes are represented as image objects 30. These objects include the criteria, conditions and constraints related to the size feature used for grouping the objects together in the depicted cluster hierarchy. In the particular example depicted in Fig. 4B, objects O16
20 62, O17 64 and O18 56 are used to form an clustering hierarchy having three hierarchal levels based on size. Object O16 62 represents the size criteria which forms the clustering hierarchy. Object O17 64 represents a second level of size criteria of less than 50 units, wherein such units may represent pixels for example; object O18 56 represents a third level of size criteria of less than 10 units. Thus, as
25 depicted in Fig. 4B, objects O11 52, O12 54, O13 56, O14 58 and O15 60 are each characterized as having a specified size of a certain number of units. Similarly, objects O13 56, O14 58 and O15 60 are each characterized as having a specified size of less than 50 units, and object O15 60 is characterized as having a specified size of less than 10 units.

30 Although Figs. 4A and 4B depict an example of a single clustering hierarchy based on only a single set of criteria, namely size, multiple clustering

hierarchies using different criteria involving multiple features may also be used for any image. For example, such clustering hierarchies may group together objects based on any combination of media, visual, and/or semantic features. This procedure is similar to the procedure used to cluster images together in visual information retrieval engines. Each object contained within the overall large image is assigned an image object 30 in the object set 24, and may also be assigned certain associated features such as media features 36, visual features 38 or semantic features 40. The intermediate nodes 32 of the object hierarchy 26 are represented as image objects 30, and also include the criteria, conditions and constraints related to one or more features used for grouping the objects together at that particular level. An image description may include any number of clustering hierarchies. The exemplary clustering hierarchy depicted in Figs. 4A and 4B is expressed in an exemplary XML implementation below.

```

15      <image>
      <image_object_set>
          <image_object type="LOCAL" id="O11"> <!-- Real objects of the image -->
              <size> <num_pixels> 120 </num_pixels> </size>
          </image_object> <!-- Others objects -->
          <image_object type="LOCAL" id="O17"> <!-- Intermediate nodes in the
20      hierarchy-->
              <size> <num_pixels> <less_than> 50 </less_than> </num_pixels>
          </size>
          </image_object> <!-- Others objects -->
      </image_object_set>
25      <object_hierarchy>
          <object_node id="ON11" object_ref="O16">
              <object_node id="ON12" object_ref="O11" />
              <object_node id="ON13" object_ref="O12" />
              <object_node id="ON14" object_ref="O17">
30              <object_node id="ON15" object_ref="O13" />
              <object_node id="ON16" object_ref="O14" />
              <object_node id="ON17" object_ref="O18">
                  <object_node id="ON18" object_ref="O15" />
              </object_node>
35              </object_node>
          </object_node>
      </object_hierarchy>
  
```

</object_hierarchy>
</image>

As depicted in the multiple clustering hierarchy example of Figs. 4A and 4B, and as denoted in Table 3 below, there are defined three levels of abstraction based on the size of the objects depicted. This multi-level abstraction scheme provides a scalable method for retrieving and viewing objects in the image depicted in Fig. 4A. Such an approach can also be used to represent multiple abstraction levels based on other features, such as various semantic classes for example.

Table 3: Objects in Each Abstraction Level

Abstraction Level	Objects
1	O11, O12
2	O11, O12, O13, O14
3	O11, O12, O13, O14, O15

Although such hierarchical structures are suitable for purposes of retrieving images, certain relationships among objects cannot adequately be expressed using such structures. Thus, as depicted in Figs. 1C and 2, the image description system of the present invention also utilizes entity-relation (E-R) graphs for the specification of more complex relationships among objects. An entity-relation graph is a graph of one or more entity nodes and the relationships among them. Table 4 below denotes several different exemplary types of such relationships, as well as specific examples of each.

Table 4: Examples of relation types and relations.

Relation Type	Relations
Spatial	
5	Directional Top Of, Bottom Of, Right Of, Left Of, Upper Left Of, Upper Right Of, Lower Left Of, Lower Right Of
	Topological Adjacent To, Neighboring To, Nearby, Within, Contain
Semantic	Relative Of, Belongs To, Part Of, Related To, Same As, Is A, Consist Of

10 Entity-relation graphs can be of any general structure, and can also be customized for any particular application by utilizing various inheritance relationships. The exemplary entity-relation graph depicted in Fig. 1C describes an exemplary spatial relationship, namely "Left Of", and an exemplary semantic relationship, namely "Shaking Hands With", between objects O1 2 and O2 6 depicted in Fig. 1A.

15 As depicted in Fig. 2, the image description system of the present invention allows for the specification of zero or more entity-relation graphs 28 (<entity_relation_graph>). An entity-relation graph 28 includes one or more sets of entity-relation elements 34 (<entity_relation>), and also contains two optional attributes, namely a unique identifier ID and a string type to describe the binding
 20 expressed by the entity relation graph 28. Values for such types could for example be provided by a thesaurus. Each entity relation element 34 contains one relation element 44 (<relation>), and may also contain one or more entity node elements 42 (<entity_node>) and one or more entity-relation elements 34. The relation element 44 contains the specific relationship being described. Each entity node element 42
 25 references an image object 30 in the image object set 24 via link 43, by utilizing an

attribute, namely object_ref. Via link 43, image objects 30 also can reference back to the entity nodes 42 referencing the image objects 30 by utilizing an attribute (event_code_refs).

As depicted in the exemplary entity-relation graph 28 of Fig. 1C, the entity-relation graph 28 contains two entity relations 34 between object O1 2 ("Person A") and object O2 6 ("Person B"). The first such entity relation 34 describes the spatial relation 44 regarding how object O1 2 is positioned with respect to (i.e., to the "Left Of") object O2 6. The second such entity relation 34 depicted in Fig. 1C describes the semantic relation 44 of how object O1 2 is "Shaking Hand With" object O2 6. An exemplary XML implementation of the entity-relation graph example depicted in Fig. 1C is shown below:

```

<entity_relation_graph>
  <entity_relation> <!-- Spatial, directional entity relation -->
    <relation type="SPATIAL.DIRECTIONAL"> Left Of </relation>
    <entity_node id="ETN1" object_ref="O1"/> <entity_node id="ETN2" object_ref="O2"/>
  </entity_relation>
  <entity_relation> <!-- Semantic entity relation -->
    <relation type="SEMANTIC"> Shaking hands with </relation>
    <entity_node id="ETN3" object_ref="O2"/> <entity_node id="ETN4"
20  object_ref="O1"/>
  </entity_relation>
</entity_relation_graph>

```

For purposes of efficiency, entity-relation elements 34 may also include one or more other entity-relation elements 34, as depicted in Fig. 2. This allows the creation of efficient nested graphs of entity relationships, such as those utilized in the Synchronized Multimedia Integration Language (SMIL), which synchronizes different media documents by using a series of nested parallel sequential relationships.

An object hierarchy 26 is a particular type of entity-relation graph 28 and therefore can be implemented using an entity-relation graph 28, wherein entities are related by containment relationships. Containment relationships are topological

relationships such as those denoted in Table 4. To illustrate that an object hierarchy 26 is a particular type of an entity-relation graph 28, the exemplary object hierarchy 26 depicted in Fig. 1B is expressed below in XML as an entity-relation graph 28.

```

5      <entity_relation_graph>
        <entity_relation>
          <relation type="SPATIAL"> Contain </relation>
          <entity_node object_ref="O0"/> <entity_node object_ref="O1"/>
        </entity_relation>
        <entity_relation>
10       <relation type="SPATIAL"> Contain </relation>
          <entity_node object_ref="O0"/> <entity_node object_ref="O2"/>
        </entity_relation>
      </entity_relation_graph>

```

The exemplary hierarchy depicted in Fig. 1B describes how object O0 8 (the overall photograph) spatially contains objects O1 2 ("Person A") and O2 6 ("Person B"). Thus, based on particular requirements, applications may implement hierarchies utilizing either the convenience of the comprehensive structure of an entity-relation graph 28, or alternatively by utilizing the efficiency of object hierarchies 26.

For image descriptors associated with any type of features, such as media features 36, visual features 38 or semantic features 40 for example, the image description system of the present invention may also contain links to extraction and similarity matching code in order to facilitate code downloading, as illustrated in the XML example below. These links provide a mechanism for efficient searching and filtering of image content from different sources using proprietary descriptors. Each image descriptor in the image description system of the present invention may include a descriptor value and a code element, which contain information regarding the extraction and similarity matching code for that particular descriptor. The code elements (<code>) may also include pointers to the executable files (<location>), as well as the description of the input parameters (<input_parameters>) and output parameters (<output_parameters>) for executing the code. Information about the type of code (namely, extraction code or similarity matching code), the code language

(such as Java or C for example), and the code version are defined as particular attributes of the code element.

The exemplary XML implementation set forth below provides a description of a so-called Tamura texture feature, as set forth in H. Tamura, S. Mori, and T. Yamawaki, "Textual Features Corresponding to Visual Perception," IEEE Transactions on Systems, Man and Cybernetics, Vol. 8, No. 6, June 1978, the entire content of which is incorporated herein by reference. The Tamura texture feature provides the specific feature values (namely, coarseness, contrast, and directionality) and also links to external code for feature extraction and similarity matching. In the feature extraction example shown below, additional information about input and output parameters is also provided. Such a description could for example be generated by a search engine in response to a texture query from a meta search engine. The meta search engine could then use the code to extract the same feature descriptor from the results received from other search engines, in order to generate a homogeneous list of results for a user. In other cases, only the extraction and similarity matching code, but not the specific feature values, is included. If necessary in such instances, filtering agents may be used to extract feature values for processing.

The exemplary XML implementation shown below also illustrates the way in which the XML language enables externally defined description schemes for descriptors to be imported and combined into the image description system of the present invention. In the below example, an external descriptor for the Croma Key shape feature is imported into the image description by using XML namespaces. Using this framework, new features, types of features, and image descriptors can be conveniently included in an extensible and modular way.

```

25      <texture> <tamura>
          <tamura_value coarseness="0.01" contrast="0.39" directionality="0.7"/>
          <code type="EXTRACTION" language="JAVA" version="1.1"> <!-- Link extraction
code -->
          <location> <location_site href="ftp://extract.tamura.java"/> </location>
          <input_parameters> <parameter name="image" type="PPM"/>
30      </input_parameters>
          <output_parameters>

```

```

                    <parameter name="tamura texture" type="double[3]"/>
                </output_parameters>
            </code>
            <code type="DISTANCE" language="JAVA" version="4.2"> <!-- Link similarity code --
5            >
                <location> <location_site href="ftp://distance.tamura.java"/> </location>
            </code>
        </tamura> </texture>

        <shape> <!-- Import external shape descriptor DTD -->
10        <chromaKeyShape xmlns:extShape "http://www.other.ds/chromaKeyShape.dtd">
            <extShape:HueRange>
                <extShape:start> 40 </extShape:start> <extShape:end> 40
            </extShape:end>
            </extShape:HueRange>
15        </chromaKeyShape>
    </shape>

```

The image description system of the present invention also supports modality transcoding. In an exemplary instance in which a content broadcaster must transmit image content to a variety of users, the broadcaster must transcode the image content into different media modalities and resolutions, in order to accommodate the users' various terminal requirements and bandwidth limitations. The image description system of the present invention provides modality transcoding in connection with both local and global objects. This modality transcoding transcodes the media modality, resolution, and location of transcoded versions of the image objects in question, or alternatively links to external transcoding code. The image descriptor in question also can point to code for transcoding an image object into different modalities and resolutions, in order to satisfy the requirements of different user terminals. The exemplary XML implementation shown below illustrates providing an audio transcoded version for an image object.

```

30    <image_object type="GLOBAL" id="O0">
        <img_obj_media_features>
            <location> <location_site href="Hi.gif"/> </location>
            <modality_transcoding>
                <modality_object_set>

```

25

```

        <modality_object id="mo2" type="AUDIO" resolution="1">
            <location><location_site
href="Hi.au.xml"?o1/></location>
        </modality_object>
5      <modality_object_set>
        </modality_transcoding>
        <img_obj_media_features>
        </image_object>

```

Fig. 5 depicts a block diagram of an exemplary computer system for
 10 implementing the image description system of the present invention. The computer
 system depicted includes a computer processor section 402 which receives digital data
 representing image content, via image input interface 404 for example. Alternatively,
 the digital image data can be transferred to the processor section 402 from a remote
 source via a bidirectional communications input/output (I/O) port 406. The image
 15 content can also be transferred to the processor section 402 from non-volatile
 computer media 408, such as any of the optical data storage or magnetic storage
 systems well known in the art. The processor section 402 provides data to an image
 display system 410, which generally includes appropriate interface circuitry and a
 high resolution monitor, such as a standard SVGA monitor and video card which are
 20 commonly employed in conventional personal computer systems and workstations for
 example. A user input device, such as a keyboard and digital pointing device a
 mouse, trackball, light pen or touch screen for example), is coupled to the processor
 section 402 to effect the user's interaction with the computer system. The exemplary
 computer system of Fig. 5 will also normally include volatile and non volatile
 25 computer memory 414, which can be accessed by the processor section 402 during
 processing operations.

Fig. 6 depicts a flow chart diagram which further illustrates the
 processing operations undertaken by the computer system depicted in Fig. 5 for
 purposes of implementing the image description system of the present invention.
 30 Digital image data 310 is applied to the computer system via link 311. The computer
 system, under the control of suitable application software, performs image object
 extraction in block 320, in which image objects 30 and associated features, such as

media features 36, visual features 38 and semantic features 40 for example, are generated. Image object extraction 320 may take the form of a fully automatic processing operation, a semi-automatic processing operation, or a substantially manual operation in which objects are defined primarily through user interaction, such as via user input device 412 for example.

In a preferred embodiment, image object extraction 320 consists of two subsidiary operations, namely image segmentation as depicted by block 325, and feature extraction and annotation as depicted by block 326. For the image segmentation 325 step, any region tracking technique which partitions digital images into regions that share one or more common characteristics may be employed. Likewise, for the feature extraction and annotation step 326, any technique which generates features from segmented regions may be employed. A region-based clustering and searching subsystem is suitable for automated image segmentation and feature extraction. An image object segmentation system is an example of a semi-automated image segmentation and feature extraction system. Manual segmentation and feature extraction could alternatively be employed. In an exemplary system, image segmentation 325 may for example generate image objects 30, and feature extraction and annotation 326 may for example generate the features associated with the image objects 30, such as media features 36, visual features 38 and semantic features 40, for example.

The object extraction processing 320 generates an image object set 24, which contains one or more image objects 30. The image objects 30 of the image object set 24 may then be provided via links 321, 322 and 324 for further processing in the form of object hierarchy construction and extraction processing as depicted in block 330, and/or entity relation graph generation processing as depicted in block 336. Preferably, object hierarchy construction and extraction 330 and entity relation graph generation 336 take place in parallel and via link 327. Alternatively, image objects 30 of the image object set 24 may be directed to bypass object hierarchy construction and extraction 330 and entity relation graph generation 336, via link 323. The object hierarchy construction and extraction 330 thus generates one or more object

hierarchies 26, and the entity relation graph generation 336 thus generates one or more entity relation graphs 28.

The processor section 402 then merges the image object set 24, object hierarchies 26 and entity relation graphs 28 into an image description record for the image content in question. The image description record may then be stored directly in database storage 340, or alternatively may first be subjected to compression by binary encoder 360 via links 342 and 361, or to encoding by description definition language encoding (using XML for example) by XML encoder 350 via links 341 and 351. Once the image description records have been stored in data base storage 340, they remain available in a useful format for access and use by other applications 370, such as search, filter and archiving applications for example, via bidirectional link 371.

Referring to Fig. 7, an exemplary embodiment of a client-server computer system on which the image description system of the present invention can be implemented is provided. The architecture of the system 100 includes a client computer 110 and a server computer 120. The server computer 120 includes a display interface 130, a query dispatcher 140, a performance database 150, query translators 160, 161, 165, target search engines 170, 171, 175, and multimedia content description systems 200, 201, 205, which will be described in further detail below.

While the following disclosure will make reference to this exemplary client-server embodiment, those skilled in the art should understand that the particular system arrangement may be modified within the scope of the invention to include numerous well-known local or distributed architectures. For example, all functionality of the client-server system could be included within a single computer, or a plurality of server computers could be utilized with shared or separated functionality.

Commercially available metasearch engines act as gateways linking users automatically and transparently to multiple text-based search engines. The system of Fig. 7 grows upon the architecture of such metasearch engines and is designed to intelligently select and interface with multiple on-line multimedia search engines by ranking their performance for different classes of user queries.

Accordingly, the query dispatcher 140, query translators 160, 161, 165, and display interface 130 of commercially available metasearch engines may be employed in the present invention.

5 The dispatcher 140 selects the target search engines to be queried by consulting the performance database 150 upon receiving a user query. This database 150 contains performance scores of past query successes and failures for each supported search option. The query dispatcher only selects search engines 170, 171, 175 that are able to satisfy the user's query, e.g. a query seeking color information will trigger color enabled search engines. Search engines 170, 171, 175 may for example
10 be arranged in a client-server relationship, such as search engine 170 and associated client 172.

The query translators 160, 161, 165, translate the user query to suitable scripts conforming to the interfaces of the selected search engines. The display component 130 uses the performance scores to merge the results from each search
15 engine, and presents them to the user.

In accordance with the present invention, in order to permit a user to intelligently search the Internet or a regional or local network for visual content, search queries may be made either by descriptions of multimedia content generated by the present invention, or by example or sketch. Each search engine 170, 171, 175
20 employs a description scheme, for example the description schemes described below, to describe the contents of multimedia information accessible by the search engine and to implement the search.

In order to implement a content-based search query for multimedia information, the dispatcher 140 will match the query description, through the
25 multimedia content description system 200, employed by each search engine 170, 171, 175 to ensure the satisfaction of the user preferences in the query. It will then select the target search engines 170, 171, 175 to be queried by consulting the performance database 150. If for example the user wants to search by color and one search engine does not support any color descriptors, it will not be useful to query that
30 particular search engine.

Next, the query translators 160, 161, 165 will adapt the query description to descriptions conforming to each selected search engine. This translation will also be based on the description schemes available from each search engine. This task may require executing extraction code for standard descriptors or downloaded
5 extraction code from specific search engines to transform descriptors. For example, if the user specifies the color feature of an object using a color coherence of 166 bins, the query translator will translate it to the specific color descriptors used by each search engine, e.g. color coherence and color histogram of x bins.

Before displaying the results to the user, the query interface will merge
10 the results from each search option by translating all the result descriptions into a homogeneous one for comparison and ranking. Again, similarity code for standard descriptors or downloaded similarity code from search engines may need to be executed. User preferences will determine how the results are displayed to the user.

Referring next to Fig. 8, a description system 200 which, in accordance
15 with the present invention, is employed by each search engine 170, 171, 175 is now described. In the preferred embodiment disclosed herein, XML is used to describe multimedia content.

The description system 200 advantageously includes several multimedia processing, analysis and annotation sub-systems 210, 220, 230, 240, 250,
20 260, 270, 280 to generate a rich variety of descriptions for a collection of multimedia items 205. Each subsystem is described in turn.

The first subsystem 210 is a region-based clustering and searching system which extracts visual features such as color, texture, motion, shape, and size for automatically segmented regions of a video sequence. The system 210
25 decomposes video into separate shots by scene change detection, which may be either abrupt or transitional (e.g. dissolve, fade in/out, wipe). For each shot, the system 210 estimates both global motion (i.e. the motion of dominant background) and camera motion, and then segments, detects, and tracks regions across the frames in the shot computing different visual features for each region. For each shot, the description
30 generated by this system is a set of regions with visual and motion features, and the camera motion. A complete description of the region-based clustering and searching

system 210 is contained in co-pending PCT Application Serial No. PCT/US98/09124, filed May 5, 1998, entitled "An Algorithm and System Architecture for Object-Oriented Content-Based Video Search," the contents of which are incorporated by reference herein.

5 As used herein, a "video clip" shall refer to a sequence of frames of video information having one or more video objects having identifiable attributes, such as, by way of example and not of limitation, a baseball player swinging a bat, a surfboard moving across the ocean, or a horse running across a prairie. A "video object" is a contiguous set of pixels that is homogeneous in one or more features of
10 interest, e.g., texture, color, motion or shape. Thus, a video object is formed by one or more video regions which exhibit consistency in at least one feature. For example a shot of a person (the person is the "object" here) walking would be segmented into a collection of adjoining regions differing in criteria such as shape, color and texture, but all the regions may exhibit consistency in their motion attribute.

15 The second subsystem 220 is an MPEG domain face detection system, which efficiently and automatically detects faces directly in the MPEG compressed domain. The human face is an important subject in images and video. It is ubiquitous in news, documentaries, movies, etc., providing key information to the viewer for the understanding of the video content. This system provides a set of regions with face
20 labels. A complete description of the system 220 is contained in PCT Application Serial No. PCT/US 97/20024, filed November 4, 1997, entitled "A Highly Efficient System for Automatic Face Region Detection in MPEG Video," the contents of which are incorporated by reference herein.

 The third subsystem 230 is a video object segmentation system in
25 which automatic segmentation is integrated with user input to track semantic objects in video sequences. For general video sources, the system allows users to define an approximate object boundary by using a tracing interface. Given the approximate object boundary, the system automatically refines the boundary and tracks the movement of the object in subsequent frames of the video. The system is robust
30 enough to handle many real-world situations that are difficult to model using existing approaches, including complex objects, fast and intermittent motion, complicated

backgrounds, multiple moving objects and partial occlusion. The description generated by this system is a set of semantic objects with the associated regions and features that can be manually annotated with text. A complete description of the system 230 is contained in U.S. Patent Application Serial No. 09/405,555, filed
5 September 24, 1998, entitled "An Active System and Algorithm for Semantic Video Object Segmentation," the contents of which are incorporated by reference herein.

The fourth subsystem 240 is a hierarchical video browsing system that parses compressed MPEG video streams to extract shot boundaries, moving objects, object features, and camera motion. It also generates a hierarchical shot-based
10 browsing interface for intuitive visualization and editing of videos. A complete description of the system 240 is contained in PCT Application Serial No. PCT/US 97/08266, filed May 16, 1997, entitled "Efficient Query and Indexing Methods for Joint Spatial/Feature Based Image Search," the contents of which is incorporated by reference herein.

15 The fifth subsystem 250 is the entry of manual text annotations. It is often desirable to integrate visual features and textual features for scene classification. For images from on-line news sources, e.g. Clarinet, there is often textual information in the form of captions or articles associated with each image. This textual information can be included in the descriptions.

20 The sixth subsystem 260 is a system for high-level semantic classification of images and video shots based on low-level visual features. The core of the system consists of various machine learning techniques such as rule induction, clustering and nearest neighbor classification. The system is being used to classify images and video scenes into high level semantic scene classes such as {nature
25 landscape}, {city/suburb}, {indoor}, and {outdoor}. The system focuses on machine learning techniques because we have found that the fixed set of rules that might work well with one corpus may not work well with another corpus, even for the same set of semantic scene classes. Since the core of the system is based on machine learning techniques, the system can be adapted to achieve high performance for different
30 corpora by training the system with examples from each corpus. The description generated by this system is a set of text annotations to indicate the scene class for each

image or each keyframe associated with the shots of a video sequence. A complete description of the system 260 is contained in S. Paek et al., "Integration of Visual and Text based Approaches for the Content Labeling and Classification of Photographs," ACM SIGIR'99 Workshop on Multimedia Indexing and Retrieval. Berkeley, C A
5 (1999), the contents of which are incorporated by reference herein.

The seventh subsystem 270 is model based image classification system. Many automatic image classification systems are based on a pre-defined set of classes in which class-specific algorithms are used to perform classification. The system 270 allows users to define their own classes and provide examples that are
10 used to automatically learn visual models. The visual models are based on automatically segmented regions, their associated visual features, and their spatial relationships. For example, the user may build a visual model of a portrait in which one person wearing a blue suit is seated on a brown sofa, and a second person is standing to the right of the seated person. The system uses a combination of lazy-
15 learning, decision trees and evolution programs during classification. The description generated by this system is a set of text annotations, i.e. the user defined classes, for each image. A complete description of the system 270 is contained in PCT Application Serial No. PCT/US 97/08266, filed May 16, 1997, entitled "A Method and Architecture for Indexing and Editing Compressed Video Over the World Wide
20 Web," the contents of which are incorporated by reference herein.

Other subsystems 280 may be added to the multimedia content description system 200, such as a subsystems from collaborators used to generate descriptions or parts of descriptions, for example.

In operation, the image and video content 205 may be a database of
25 still images or moving video, a buffer receiving content from a browser interface 206, or a receptacle for live image or video transmission. The subsystems 210, 220, 230, 240, 250, 260, 270, 280 operate on the image and video content 205 to generate descriptions 211, 221, 231, 241, 251, 261, 271, 281 that include low level visual features of automatically segmented regions, user defined semantic objects, high level
30 scene properties, classifications and associated textual information, as described above. Once all the descriptions for an image or video item are generated and

integrated in block 290, the descriptions are then input into a database 295, which the search engine 170 accesses.

It should be noted that certain of the subsystems, i.e., the region-based clustering and searching subsystem 210 and the video object segmentation system 230 may implement the entire description generation process, while the remaining subsystems implement only portions of the process and may be called on by the subsystems 210, 230 during processing. In a similar manner, the subsystems 210 and 230 may be called on by each other for specific tasks in the process.

In Figures 1-6, systems and methods for describing image content are described. These techniques are readily extensible to video content as well. The performance of systems for searching and processing video content information can benefit from the creation and adoption of a standard by which such video content can be thoroughly and efficiently described. As used herein, the term "video clip" refers to an arbitrary duration of video content, such as a sequence of frames of video information. The term description scheme refers to the data structure or organization used to describe the video content. The term description record refers to the description scheme wherein the data fields of the data structure are defined by data which describes the content of a particular video clip.

Referring to Figure 9, an exemplary embodiment of the present video description scheme (DS) is illustrated in schematic form. The video DS inherits all of the elements of the image description scheme and adds temporal elements thereto, which are particular to video content. Thus, a video element 922 which represents a video description, generally includes a video object set 924, an object hierarchy definition 926 and entity relation graphs 928, all of which are similar to those described in connection with Figure 2. An exemplary video DS definition is illustrated below in Table 5.

Table 5: Elements in the Video Description Scheme (DS).

Element	Contains	May be Contained in
video	video_object_set (1) object_hierarchy ¹ (0..*)	(root element)

	entity_relation_graph ¹ (0..*)	
video_object_set	video_object (1..*)	video
video_object	vid_obj_media_features (0..1) vid_obj_semantic_features (0..1) vid_obj_visual_features (0..1) vid_obj_temporal_features (0..1)	object_set ¹
vid_obj_media_features	location ¹ (0..1) file_format ¹ (0..1) file_size ¹ (0..1) resolution ¹ (0..1) modality_transcoding ¹ (0..1) bit_rate (0..1)	video_object
vid_obj_semantic_features	text_annotation ¹ (0..1) who, what_object, what_action, when, where, why (0..1)	video_object
vid_obj_visual_features & type="LOCAL"	image_scl ¹ (0..1) color ¹ (0..1) texture ¹ (0..1) shape ¹ (0..1) size ¹ (0..1) position ¹ (0..1) motion (0..1)	video_object
vid_obj_visual_features & (type="SEGMENT" type="GLOBAL")	video_scl (0..1) visual_sprite (0..1) transition (0..1) camera_motion (0..1) size(0..1) key_frame (0..*)	video_object
vid_obj_visual_features	time (0..1)	video_object
object_hierarchy ¹	object_node ¹ (1)	video
object_node ¹	object_node ¹ (0..*)	object_hierarchy ¹ object_node ¹

5

10

entity_relation_graph ¹	entity_relation ¹ (1..*)	Video
entity_relation ¹	relation ¹ (1)	Entity_relation_graph ¹
	entity_node ¹ (1..*)	Entity_relation ¹
	entity_relation ¹ (0..*)	

¹ Defined in the Image DTD7

A basic element of the present video description scheme (DS) is the video object (<video_object>) 930. A video object 930 refers to one or more arbitrary regions in one or more frames of a video clip. For example, and not by way of limitation, a video object may be defined as local objects, segment objects and global objects. Local objects refer to a group of pixels found in one or more frames . Segment objects refer to one or more related frames of the video clip. Global objects refer to the entire video clip.

A video object 930 is an element of the video object set 924 and can be related to other objects in the object set 924 by the object hierarchy 926 and entity relation graphs 928 in the same manner as described in connection with Figs. 1-6. Again, the fundamental difference between the video description scheme and the previously described image description scheme resides in the inclusion of temporal parameters which will further define the video objects and their interrelation in the description scheme.

In using XML to implement the present video description scheme, to indicate if a video object has associated semantic information, the video object can include a "semantic" attribute which can take on an indicative value, such as true or false. To indicate if the object has associated physical information (such as color, shape, time, motion and position), the object can include an optional "physical" attribute that can take on an indicative value, such as true or false. To indicate whether regions of an object are spatially adjacent to one another (continuous in space), the object can include an optional "spaceContinuous" attribute that can assume a value such as true or false. To indicate if the video frames which contain a particular object are temporally adjacent to one another (continuous in time), the

object can further include an optional "timeContinuous" attribute. This attribute can assume an indicative value, such as true or false. To distinguish if the object refers to a region within select frames of a video, to entire frames of the video, or to the entire video commonly (e.g. shots, scenes, stories), the object will generally include an attribute (type), that can have multiple indicative values such as, LOCAL,
 5 SEGMENT, and GLOBAL, respectively.

Figure 10 is a pictorial diagram which depicts a video clip from a video clip wherein a number of exemplary objects are identified. Object O0, is a global object which refers to the entire video clip. Object O1, the library, refers to an entire
 10 frame of video and would be classified as a segment type object. Objects O2 and O3 are local objects which refer to narrator A and narrator B, respectively, which are person objects that are continuous in time and space. Object O4 ("Narrators") are local video objects (O2, O3) which is discontinuous in space. Figure 10 further illustrates that objects can be nested. For example, object O1, the library, includes
 15 local object O2, and both of these objects are contained within the global object O0. An XML description of the objects defined in Figure 10 is set forth below.

```

    <video_object_set>
        <video_object id="O0" type="GLOBAL"> <video_object> <!--Documentary -->
        <video_object id="O1" type="SEGMENT"> </video_object> <!--Library -->
    20  <video_object id="O2" type="LOCAL"> </video_object> <!-- Narrator A -->
        <video_object id="O3" type="LOCAL"> </video_object> <!--Narrator B -->
        <video_object id="O4" type="LOCAL"> </video_object> <!--Narrators -->
    </video_object_set>
  
```

Figure 11 illustrates how two or more video objects are related through
 25 the object hierarchy 926. In this case, objects O2 and O3 have a common semantic feature of "what object" being a narrator. Thus, these objects can be referenced in the definition of a new object, O4, narrators, via an object hierarchy definition. The details of such hierarchical definition follow that described in connection with Fig. 3A.

Figure 12 illustrates how the entity relation graph in the video description scheme can relate video objects. In this case, two relationships are shown between objects O2 and O3. The first is a semantic relationship, "colleague of", which is equivalent to the type of semantic relationship which could be present in the case of the image description scheme, as described in connection with Figure 1C.. Figure 12 further shows a temporal relationship between the objects O2 and O3. In this case, object O2 precedes object O3 in time within the video clip, thus the temporal relationship "before" can be applied. In addition to the exemplary relation types and relations set forth in connection with the image description scheme, the video description scheme can employ the relation types and relations set forth in the table below.

RELATION TYPE	RELATIONS
TEMPORAL- Directional	Before, After, Immediately Before, Immediately After
TEMPORAL-Topological	Co-Begin, Co-End, Parallel, Sequential, Overlap, Within, Contain, Nearby

The video objects 930 can be further characterized in terms of object features. Although any number and type of features can be defined to characterize the video objects in a modular and extensible manner, a useful exemplary feature set can include semantic features 940, visual features 938, media features 936 and temporal features 937. Each feature can then be further defined by feature parameters, or descriptors. Such descriptors will generally follow that described in connection with the image description scheme, with the addition of requisite temporal information. For example, visual features 938 can include a set of descriptors such as shape, color, texture, and position, as well as motion parameters. Temporal features 937 will generally include such descriptors as start time, end time and duration. Table 6 shows examples of descriptors, in addition to those set forth in connection with the image description scheme, that can belong to each of these exemplary classes of features.

Table 6: Feature classes and features.

Feature Class	Features
Visual	Motion, Editing effect, Camera Motion
Temporal	Start Time, End Time, Duration

5 In summary, in a fashion analogous to the previously described image description scheme, the present video description scheme includes a video object set 924, an object hierarchy 926, and entity relation graphs 928. Video objects 930 are further defined by features. The objects 930 within the object set 924 can be related hierarchically by one or more object hierarchy nodes 932 and references 933.

10 Relations between objects 930 can also be expressed in entity relation graphs 928, which further include entity relations 934, entity nodes 942, references 943 and relations 944, all of which substantially correspond in the manner described in connection with Figure 2. Each video object 930 preferably includes features that can link to external extraction and similarity matching code.

15 Figure 13 is a block diagram of an exemplary computer system for implementing the present video description systems and methods, which is analogous to the system described in connection with Figure 5. The system includes a computer processor section 1302 which receives digital data representing video content, such as via video input interface 1304. Alternatively, the digital video data can be transferred
20 to the processor from a remote source via a bidirectional communications input/output port 1306. The video content can also be transferred to the processor section 1302 from computer accessible media 408, such as optical data storage systems or magnetic storage systems which are known in the art. The processor section 1302 provides data to a video display system 1310, which generally includes appropriate interface
25 circuitry and a high resolution monitor, such as a standard SVGA monitor and video card commonly employed in conventional personal computer systems and workstations. A user input device 1312, such as a keyboard and digital pointing device, such as a mouse, trackball, light pen, touch screen and the like, is operatively coupled to the processor section 1302 to effect user interaction with the system. The

system will also generally include volatile and non volatile computer memory 1314 which can be accessed by the processor section during processing operations.

Figure 14 is a flow diagram which generally illustrates the processing operations undertaken by processor section 1302 in establishing the video DS
5 described in connection with Figures 9-12. Digital data representing a video clip is applied to the system, such as via video input interface 1304 and is coupled to the processor section 1302. The processor section 1302, under the control of suitable software, performs video object extraction processing 1402 wherein video objects 930, features 936, 937, 938, 940 and the associated descriptors are generated. Video
10 object extraction processing 1402 can take the form of a fully automatic processing operation, a semi-automatic processing operation, or a substantially manual operation where objects are largely defined through user interaction via the user input device 1312.

The result of object extraction processing is the generation of an object
15 set 924, which contains one or more video objects 930 and associated object features 936, 937, 938, 940. The video objects 930 of the object set 924 are subjected to further processing in the form of object hierarchy construction and extraction processing 1404 and entity relation graph generation processing 1406. Preferably, these processing operations take place in a parallel fashion. The output result from
20 object hierarchy construction and extraction processing 1404 is an object hierarchy 926. The output result of entity relation graph generation processing 1406 is one or more entity relation graphs 928. The processor section 1302 combines the object set, object hierarchy and entity relation graphs into a description record in accordance with the present video description scheme for the applied video content. The
25 description record can be stored in database storage 1410, subjected to low-level encoding 1412 (such as binary coding) or subjected to description language encoding (e.g. XML) 1414. Once the description records are stored in the read/write storage 1308 in the form of a database, the data is available in a useful format for use by other applications 1416, such as search, filter, archiving applications and the like.

Exemplary Document Type Definition of Video Description scheme

This section discusses one embodiment wherein XML has been used to implement a document type definition (DTD) of the present video description scheme. Table 1, set forth above, summarizes the DTD of the present video DS. Appendix A includes the full listing of the DTD of the video DS. In general, a Document Type Definition (DTD) provides a list of the elements, tags, attributes, and entities contained in the document, and their relationships to each other. In other words, DTDs specify a set of rules for the structure of a document. DTDs may be included in a computer data file that contains the document they describe, or they may be linked to or from an external universal resource location (URL). Such external DTDs can be shared by different documents and Web sites. A DTD is generally included in a document's prolog after the XML declaration and before the actual document data begins.

Every tag used in a valid XML document must be declared exactly once in the DTD with an element type declaration. The first element in a DTD is the root tag. In our video DS, the root tag can be designated as <video> tag. An element type declaration specifies the name of a tag, the allowed children of the tag, and whether the tag is empty. The root <video> tag can be defined, as follows:

<!ELEMENT video (video_object_set, object_hierarchy*, entity_relation_graph*)>

where the asterisk (*) indicates zero or more occurrences. In XML syntax, the plus sign (+) indicates one or more occurrences and the question mark (?) indicates zero or one occurrence.

In XML, all element type declarations start with <!ELEMENT and end with >. They include the name of the tag being declared video and the allowed contents (video_object_set, object_hierarchy*, entity_relation_graph*). This declaration indicates that a video element must contain a video object set element (<video_object_set>), zero or more object hierarchy elements (<object_hierarchy>), and zero or more entity relation graph elements (<entity_relation_graph>).

The video object set 924 can be defined as follows.

```

    <!-- Video object set element -->
    <!-- An object set consists of one or more video objects -->
    <! ELEMENT video_object_set (video_object+)>
    <!-- Video object element -->
5    <!-- Video object elements consist of the following elements:
    - optional vid_obj_media_features element
    - optional vid_obj_semantic_features element
    - optional vid_obj_visual_features element
    - optional vid_obj_temporal_features element -->
10    <!ELEMENT video_object (vid_obj_media_features?,
    vid_obj_semantic_features?,
        vid_obj_visual_features?,
    vid_obj_temporal_features?)>
    <!-- Video object elements must have a unique ID attribute in each description -->
15    <!ATTLIST video_object
        type (LOCAL|SEGMENT|GLOBAL) #REQUIRED
        id ID #IMPLIED
        object_ref IDREF #IMPLIED
        object_node_ref IDREFS #IMPLIED
20        entity_node_ref IDREFS #IMPLIED>
    <!-- Feature elements: media, semantic, temporal and visual -->
    <!-- Video object media features element consists an optional location,
    file_format, file_size, resolution, modality_transcoding, and bit_rate elements
    -->
25    <!ELEMENT vid_obj_media_features (location?, file_format?, file_size?,
    resolution?,
        modality_transcoding?, bit_rate?)>
    <!-- Video object semantic features element consists an optional text_annotation and
    the 6-W elements -->
30    <!ELEMENT vid_obj_semantic_features (text_annotation?, who?, what_action?,
        where?, why?, when?)>

```

<!-- Video object visual features element consists image_scl, color, texture, shape,
 size, position, motion, video_scl, visual_sprite, transition, and camera_motion
 elements, and multiple key_frame elements-->
 <!ELEMENT vid_obj_visual_features (image_scl?, color?, texture?, shape?, size?,
 5 position?, motion?, video_scl?, visual_sprite?, transition?, camera_motion?,
 key_frame*)>

In the above example, the first declaration indicates that a video object
 set element (<video_object_set>) 924 contains one or more video objects
 (<video_object>) 930. The second declaration indicates that a video object 930
 10 contains an optional video object media feature (<vid_obj_media_features>) 936,
 semantic feature (<vid_obj_semantic_features>) 940, visual feature
 (<vid_obj_visual_features>) 938, and temporal feature
 (<vid_obj_temporal_features>) 937 elements. In addition, the video object tag is
 defined as having one required attribute, type, that can only have three possible values
 15 (LOCAL, SEGMENT, GLOBAL); and three optional attributes, id, object_ref, and
 object_node_ref, of type ID, IDREFS, and IDREFS, respectively.

Some XML tags include attributes. Attributes are intended for extra
 information associated with an element (like an ID). The last four declarations in the
 example shown above corresponds to the video object media feature 936, semantic
 20 feature 940, visual feature 938, and temporal feature 937 elements. These elements
 group feature elements depending on the information they provide. For example, the
 media features element (<vid_obj_media_features>) 936 contains an optional
 location, file_format, file_size, resolution, modality_transcoding, and bit_rate element
 to define the descriptors of the media features 936. The semantic feature element
 25 (<vid_obj_semantic_features>) contains an optional text_annotation and the 6-W
 elements corresponding to the semantic feature descriptors 940. The visual feature
 element (<vid_obj_visual_features>) contains optional image_scl, color, texture,
 shape, size, position, video_scl, visual_sprite, transition, camera_motion elements,
 and multiple key_frame elements for the visual feature descriptors. The temporal

features element (<vid_obj_temporal_features>) contains an optional time element as the temporal feature descriptor.

In the exemplary DTD listed in Appendix A, for the sake of clarity and flexibility feature elements are declared in external DTDs using entities. The following description sets forth a preferred method of referencing a separate external DTD for each one of these elements.

In the simplest case, DTDs include all the tags used in a document. This technique becomes unwieldy with longer documents. Furthermore, it may be desirable to use different parts of a DTD in many different places. External DTDs enable large DTDs to be built from smaller ones. That is, one DTD may link to another and in so doing pull in the elements and entities declared in the first. Smaller DTD's are easier to analyze. DTDs are connected with external parameter references, as illustrated in the example following:

```

15      <!ENTITY % camera_motion PUBLIC
      "http://www.ee.columbia.edu/mpeg7/xml/features/camera_motion.dtd">
      %camera_motion;
```

The object hierarchy can be defined in the image DTD. The following example provides an overview of a declaration for the present object hierarchy element.

```

20      <!-- Object hierarchy element -->
      <!-- A hierarchy element consists of one root node -->
      <!ELEMENT object_hierarchy (object_node)>
      <!-- The object hierarchy element has two optional attributes: an id and a type -->
      <!ATTLIST object_hierarchy
25          id ID #IMPLIED
          type CDATA #IMPLIED>
      <!-- Object node element -->
      <!-- Object node elements consist of zero or more object node elements -->
      <!ELEMENT object_node (object_node*)>
30      <!-- Object node elements must have an id attribute of type ID. -->
```

```

<!ATTLIST object_node
    id ID #IMPLIED
    object_ref IDREF #REQUIRED>

```

5 The object hierarchy element (<object_hierarchy>) preferably contains a
 single root object node element (<object_node>). An object node element generally
 contains zero or more object node elements. Each object node element can have an
 associated unique identifier, id. The identifier is expressed as an optional attribute of
 the elements of type ID, e.g. <object_node id="o1" object_ref="o1">. Each object
 node element can also include a reference to a video object element by using the
 10 unique identifier associated with each video object. The reference to the video object
 element is given as an attribute of type IDREF (object_ref). Object elements can link
 back to those object node elements pointing at them by using an attribute of type
 IDREFS (object_node_ref).

15 The entity relation graph definition is very similar to the object hierarchy's
 one. An example, is listed below.

```

<!-- Entity relation graph element -->
<!-- A entity relation graph element consists of zero or more entity relation elements
-->
<!ELEMENT entity_relation_graph (entity_relation+)>
20 <!-- A entity relation graph element can include two attributes: an id and a type -->
<!-- Possible types of entity relation graphs and entity relations follow:
    - Spatial: topological, directional
    - Temporal: topological, directional
    - Semantic -->
25 <!ATTLIST entity_relation_graph
    id ID #IMPLIED
    type CDATA #IMPLIED>
<!-- Entity relation element -->
<!-- A entity relation graph element consists of one relation, and zero or more entity
30 nodes or entity relation elements -->

```

```
<!ELEMENT entity_relation (relation, (entity_node | entity_node_set |
entity_relation)*)>
```

```
<!-- A entity relation element can include a type attribute -->
```

```
<!ATTLIST entity_relation
5         type CDATA #IMPLIED>
```

```
<!-- Entity relation element -->
```

```
<!-- Examples of relations are
```

- SPATIAL.TOPOLOGICAL: overlap, etc.
- SPATIAL.DIRECTIONAL: to the left, to the right, etc.
- 10 - TEMPORAL.TOPOLOGICAL: at the same time, etc.
- TEMPORAL.DIRECTIONAL: before, after, immediately before, etc.
- SEMANTIC: father of, etc. -->

```
<!ELEMENT relation (#PCDATA | code)*>
```

```
<!-- Entity node element -->
```

```
15 <!-- This element can contain string data. It can have a unique attribute (id), and
must include a reference attribute to an object element (object_ref) -->
```

```
<!ELEMENT entity_node (#PCDATA)>
```

```
<!ATTLIST entity_node
```

```
id ID #IMPLIED
```

```
20 object_ref IDREF #REQUIRED>
```

```
<!-- Entity node set element -->
```

```
<!ELEMENT entity_node_set (entity_node+)>
```

The declaration of the entity node element can contain either one or another element by separating the child elements with a vertical bar rather than a

25 comma.

The description above sets forth a data structure of a video description scheme, as well as systems and methods of characterizing video content in accordance with the present video description scheme. Of course, the present video description scheme can be used advantageously in connection with the systems described in

30 connection with Figures 7 and 8.

Although the present invention has been described in connection with specific exemplary embodiments, it should be understood that various changes, substitutions and alterations can be made to the disclosed embodiments without departing from the spirit and scope of the invention as set forth in the appended
5 claims.

CLAIMS

1. A system for generating a description record from video information, comprising:

5 at least one video input interface for receiving said video information;

a computer processor coupled to said at least one video input interface for receiving said video information therefrom, processing said video information by performing video object extraction processing to generate video object descriptions from said video information, processing said generated video object
10 descriptions by object hierarchy construction and extraction processing to generate video object hierarchy descriptions, and processing said generated video object descriptions by entity relation graph generation processing to generate entity relation graph descriptions, wherein at least one description record including said video object descriptions, said video object hierarchy descriptions and said entity relation graph
15 descriptions is generated to represent content embedded within said video information; and

a data storage system, operatively coupled to said processor, for storing said at least one description record.

2. The system of claim 1, wherein said video object extraction
20 processing and said object hierarchy construction and extraction processing are performed in parallel.

3. The system of claim 1, wherein said video object extraction processing comprises:

video segmentation processing to segment each video in said video information into regions within said video; and

feature extraction and annotation processing to generate one or more feature descriptions for one or more said regions;

5 whereby said generated video object descriptions comprise said one or more feature descriptions for one or more said regions.

4. The system of claim 3, wherein said regions are selected from the group consisting of local, segment and global regions.

10 5. The system of claim 3, wherein said one or more feature descriptions are selected from the group consisting of media features, visual features, temporal features, and semantic features.

6. The system of claim 5, wherein said semantic features are further defined by at least one feature description selected from the group consisting
15 of who, what object, what action, where, when, why, and text annotation.

7. The system of claim 5, wherein said visual features are further defined by at least one feature description selected from the group consisting of color, texture, position, size, shape, motion, camera motion, editing effect, and orientation.

20 8. The system of claim 5, wherein said media features are further defined by at least one feature description selected from the group consisting

of file format, file size, color representation, resolution, data file location, author, creation, scalable layer and modality transcoding.

9. The system of claim 5, wherein said temporal features are further defined by at least one feature description selected from the group consisting of start time, end time and duration.

10. The system of claim 1, wherein said object hierarchy construction and extraction processing generates video object hierarchy descriptions of said video object descriptions based on visual feature relationships of video objects represented by said video object descriptions.

11. The system of claim 1, wherein said object hierarchy construction and extraction processing generates video object hierarchy descriptions of said video object descriptions based on semantic feature relationships of video objects represented by said video object descriptions.

12. The system of claim 1, wherein said object hierarchy construction and extraction processing generates video object hierarchy descriptions of said video object descriptions based on media feature relationships of video objects represented by said video object descriptions.

13. The system of claim 1, wherein said object hierarchy construction and extraction processing generates video object hierarchy descriptions of said video object descriptions based on relationships of video objects represented by said video object descriptions, wherein said relationships are selected from the

group consisting of visual feature relationships, semantic feature relationships, temporal feature relationships and media feature relationships.

14. The system of claim 1, wherein said object hierarchy construction and extraction processing generates video object hierarchy descriptions of said video object descriptions based on relationships of video objects represented by said video object descriptions, wherein said video object hierarchy descriptions have a plurality of hierarchical levels.

15. The system of claim 14, wherein said video object hierarchy descriptions having a plurality of hierarchical levels comprise clustering hierarchies.

16. The system of claim 15, wherein said clustering hierarchies are based on relationships of video objects represented by said video object descriptions, wherein said relationships are selected from a group consisting of visual feature relationships, semantic feature relationships, temporal relationships and media feature relationships.

17. The system of claim 15, wherein said video object hierarchy descriptions having a plurality of hierarchical levels are configured to comprise multiple levels of abstraction hierarchies.

18. The method of claim 17, wherein said multiple levels of abstraction hierarchies are configured to be based on relationships of video objects represented by said video object descriptions, wherein said relationships are selected from a group consisting of visual feature relationships, semantic feature relationships, temporal feature relationships and media feature relationships.

19. The system of claim 1, wherein said entity relation graph generation processing generates entity relation graph descriptions of said video object descriptions based on relationships of video objects represented by said video object descriptions, wherein said relationships are selected from the group consisting of
5 visual feature relationships, semantic feature relationships, temporal feature relationships and media feature relationships.
20. The system of claim 1, further comprising an encoder for receiving and encoding said video object descriptions into encoded description information, wherein said data storage system is operative to store said encoded
10 description information as said at least one description record.
21. The system of claim 1, wherein said video object descriptions, said video object hierarchy descriptions, and said entity relation graph descriptions are combined together to form video descriptions, and further comprising an encoder for receiving and encoding said video descriptions into encoded
15 description information, wherein said data storage system is operative to store said encoded description information as said at least one description record.
22. The system of claim 21, wherein said encoder comprises a binary encoder.
23. The system of claim 21, wherein said encoder comprises an
20 XML encoder.

24. The system of claim 1, further comprising:

a video display device operatively coupled to the computer processor for displaying the video information; and

5 at least one user input device operatively coupled to the computer processor, wherein at least a portion of said video object processing includes receiving a user input through manipulation of said user input device.

25. A method for generating a description record from video information, comprising the steps of:

receiving said video information;

10 processing said video information by performing video object extraction processing to generate video object descriptions from said video information;

processing said generated video object descriptions by object hierarchy construction and extraction processing to generate video object hierarchy
15 descriptions;

processing said generated video object descriptions by entity relation graph generation processing to generate entity relation graph descriptions, wherein at least one description record including said video object descriptions, said video object hierarchy descriptions and said entity relation graph descriptions is
20 generated to represent content embedded within said video information; and

storing said at least one description record.

26. The method of claim 25, wherein said steps of video object extraction processing and object hierarchy construction and extraction processing are performed in parallel.

27. The method of claim 25, wherein said step of video object extraction processing comprises the further steps of:

video segmentation processing to segment each video in said video information into regions within said video; and

5 feature extraction and annotation processing to generate one or more feature descriptions for one or more said regions;

whereby said generated video object descriptions comprise said one or more feature descriptions for one or more said regions.

28. The method of claim 27, wherein said regions are selected
10 from the group consisting of local, segment and global regions.

29. The method of claim 27, further comprising the step of selecting said one or more feature descriptions from the group consisting of media features, visual features, temporal and semantic features.

30. The method of claim 29, wherein said semantic features are
15 further defined by at least one feature description selected from the group consisting of who, what object, what action, where, when, why and text annotation.

31. The method of claim 29, wherein said visual features are further defined by at least one feature description selected from the group consisting of color, texture, position, size, shape, motion, editing effect, camera motion and
20 orientation.

32. The method of claim 29, wherein said media features are further defined by at least one feature description selected from the group consisting of file format, file size, color representation, resolution, data file location, author, creation, scalable layer and modality transcoding.

5 33. The method of claim 29, wherein said temporal features are further defined by at least one feature description selected from the group consisting of start time, end time and duration.

34. The method of claim 25, wherein said step of object hierarchy construction and extraction processing generates video object hierarchy
10 descriptions of said video object descriptions based on visual feature relationships of video objects represented by said video object descriptions.

35. The method of claim 25, wherein said step of object hierarchy construction and extraction processing generates video object hierarchy
15 descriptions of said video object descriptions based on semantic feature relationships of video objects represented by said video object descriptions.

36. The method of claim 25, wherein said step of object hierarchy construction and extraction processing generates video object hierarchy
descriptions of said video object descriptions based on media feature relationships of
video objects represented by said video object descriptions.

20 37. The method of claim 25, wherein said step of object hierarchy construction and extraction processing generates video object hierarchy

descriptions of said video object descriptions based on temporal feature relationships of video objects represented by said video object descriptions.

38. The method of claim 25, wherein said step of object hierarchy construction and extraction processing generates video object hierarchy
5 descriptions of said video object descriptions based on relationships of video objects represented by said video object descriptions, wherein said relationships are selected from the group consisting of visual feature relationships, semantic feature relationships, temporal feature relationships and media feature relationships.

39. The method of claim 25, wherein said step of object
10 hierarchy construction and extraction processing generates video object hierarchy descriptions of said video object descriptions based on relationships of video objects represented by said video object descriptions, wherein said video object hierarchy descriptions are configured to have a plurality of hierarchical levels.

40. The method of claim 39, wherein said video object hierarchy
15 descriptions having a plurality of hierarchical levels are configured to comprise clustering hierarchies.

41. The method of claim 40, wherein said clustering hierarchies are configured to be based on relationships of video objects represented by said video object descriptions, wherein said relationships are selected from a group consisting of
20 visual feature relationships, semantic feature relationships, temporal feature relationships and media feature relationships.

42. The method of claim 40, wherein said video object hierarchy descriptions having a plurality of hierarchical levels are configured to comprise multiple levels of abstraction hierarchies.

5 43. The method of claim 40, wherein said multiple levels of abstraction hierarchies are configured to be based on relationships of video objects represented by said video object descriptions, wherein said relationships are selected from a group consisting of visual feature relationships, semantic feature relationships, temporal feature relationships and media feature relationships.

10 44. The method of claim 25, wherein said step of entity relation graph generation processing generates entity relation graph descriptions of said video object descriptions based on relationships of video objects represented by said video object descriptions, wherein said relationships are selected from the group consisting of visual feature relationships, semantic feature relationships, temporal feature relationships and media feature relationships.

15 45. The method of claim 25, further comprising the steps of receiving and encoding said video object descriptions into encoded description information, and storing said encoded description information as said at least one description record.

20 46. The method of claim 25, further comprising the steps of combining said video object descriptions, said video object hierarchy descriptions, and said entity relation graph descriptions to form video descriptions, and receiving and encoding said video descriptions into encoded description information, and storing said encoded description information as said at least one description record.

47. The method of claim 46, wherein said step of encoding comprises the step of binary encoding.

48. The method of claim 46, wherein said step of encoding comprises the step of XML encoding.

5 49. A computer readable media containing digital information with at least one description record representing video content embedded within corresponding video information, the at least one description record comprising:

one or more video object descriptions generated from said video information using video object extraction processing;

10 one or more video object hierarchy descriptions generated from said generated video object descriptions using object hierarchy construction and extraction processing; and

one or more entity relation graph descriptions generated from said generated video object descriptions using entity relation graph generation processing.

15 50. The computer readable media of claim 49, wherein said video object descriptions, said video object hierarchy descriptions, and said entity relation graph descriptions further comprise one or more feature descriptions.

51. The computer readable media of claim 50, wherein said one or more feature descriptions are selected from the group consisting of media features,
20 visual features, temporal features and semantic features.

52. The computer readable media of claim 51, wherein said semantic features are further defined by at least one feature description selected from the group consisting of who, what object, what action, where, when, why and text annotation.

5 53. The computer readable media of claim 51, wherein said visual features are further defined by at least one feature description selected from the group consisting of color, texture, position, size, shape, motion, camera motion, editing effect and orientation.

10 54. The computer readable media of claim 51, wherein said media features are further defined by at least one feature description selected from the group consisting of file format, file size, color representation, resolution, data file location, author, creation, scalable layer and modality transcoding.

15 55. The computer readable media of claim 51, wherein said temporal features are further defined by at least one feature description selected from the group consisting of start time, end time and duration.

56. The computer readable media of claim 49, wherein said object hierarchy descriptions are based on visual feature relationships of video objects represented by said video object descriptions.

57. The computer readable media of claim 49, wherein said video object hierarchy descriptions are based on semantic feature relationships of video objects represented by said video object descriptions.

58. The computer readable media of claim 49, wherein said
5 video object hierarchy descriptions are based on media feature relationships of video objects represented by said video object descriptions.

59. The computer readable media of claim 49, wherein said video object hierarchy descriptions are based on temporal feature relationships of video objects represented by said video object descriptions.

60. The computer readable media of claim 49, wherein said
10 video object hierarchy descriptions are based on relationships of video objects represented by said video object descriptions, wherein said video object hierarchy descriptions have a plurality of hierarchal levels.

61. The computer readable media of claim 60, wherein said
15 video object hierarchy descriptions having a plurality of hierarchal levels comprise clustering hierarchies.

62. The computer readable media of claim 61, wherein said
clustering hierarchies are based on relationships of video objects represented by said
video object descriptions, wherein said relationships are selected from a group
20 consisting of visual feature relationships, semantic feature relationships, temporal
feature relationships and media feature relationships.

63. The computer readable media of claim 62, wherein said video object hierarchy descriptions having a plurality of hierarchical levels are configured to comprise multiple levels of abstraction hierarchies.

5 64. The computer readable media of claim 63, wherein said multiple levels of abstraction hierarchies are configured to be based on relationships of video objects represented by said video object descriptions, wherein said relationships are selected from a group consisting of visual feature relationships, semantic feature relationships, temporal feature relationships and media feature relationships.

10 65. The computer readable media of claim 49, wherein said entity relation graph descriptions are based on relationships of video objects represented by said video object descriptions, wherein said relationships are selected from the group consisting of visual feature relationships, semantic feature relationships, temporal feature relationships and media feature relationships.

15 66. The computer readable media of claim 49, wherein said video object descriptions are in the form of encoded description information.

20 67. The computer readable media of claim 49, wherein said video object descriptions, said video object hierarchy descriptions, and said entity relation graph descriptions are combined together in the form of encoded description information.

68. The computer readable media of claim 67, wherein said encoded description information is in the form of binary encoded information.

69. The computer readable media of claim 67, wherein said encoded description information is in the form of XML encoded information.

5 70. The system of claim 1, wherein feature descriptions include pointers to extraction and matching code to facilitate code downloading.

71. The system of claim 5, wherein feature descriptions include pointers to extraction and matching code to facilitate code downloading.

10 72. The method of claim 25, wherein feature descriptions include pointers to extraction and matching code to facilitate code downloading.

73. The method of claim 29, wherein feature descriptions include pointers to extraction and matching code to facilitate code downloading.

15 74. The computer readable media of claim 49, wherein feature descriptions include pointers to extraction and matching code to facilitate code downloading.

75. The computer readable media of claim 53, wherein feature descriptions include pointers to extraction and matching code to facilitate code downloading.

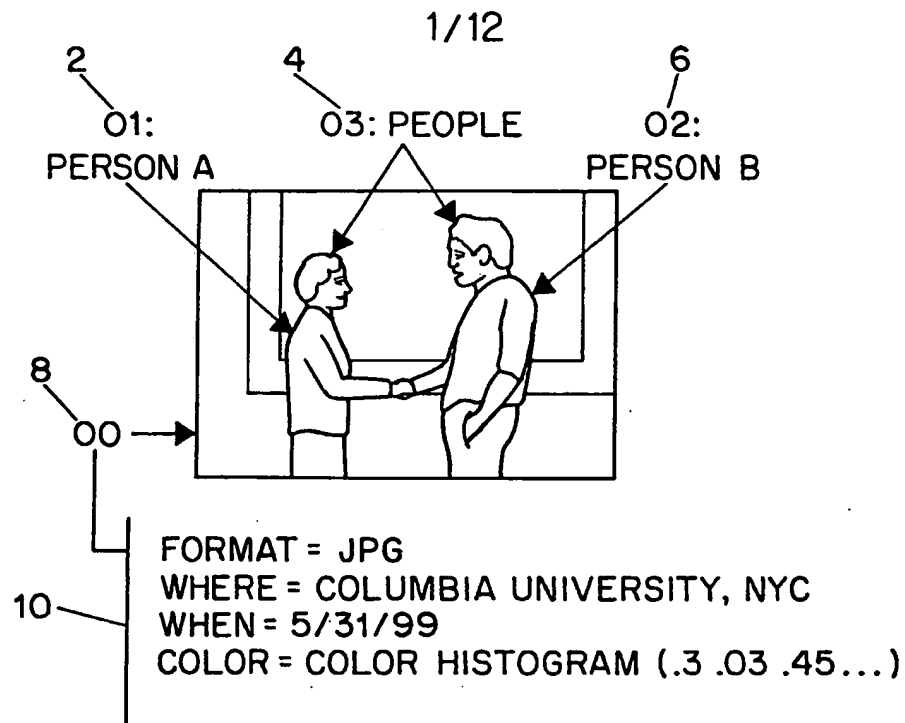


FIG. 1a

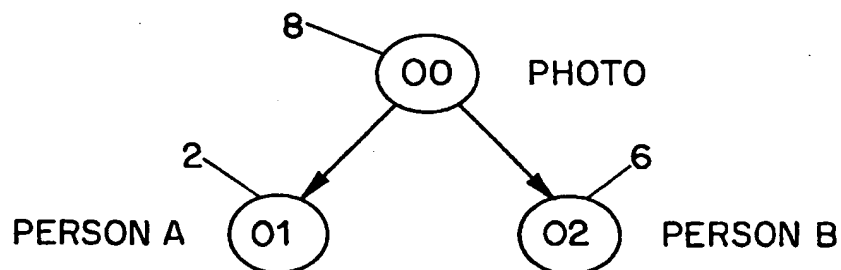


FIG. 1b

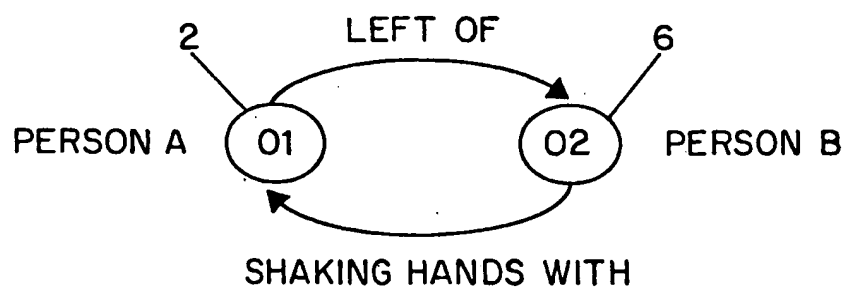


FIG. 1c

2/12

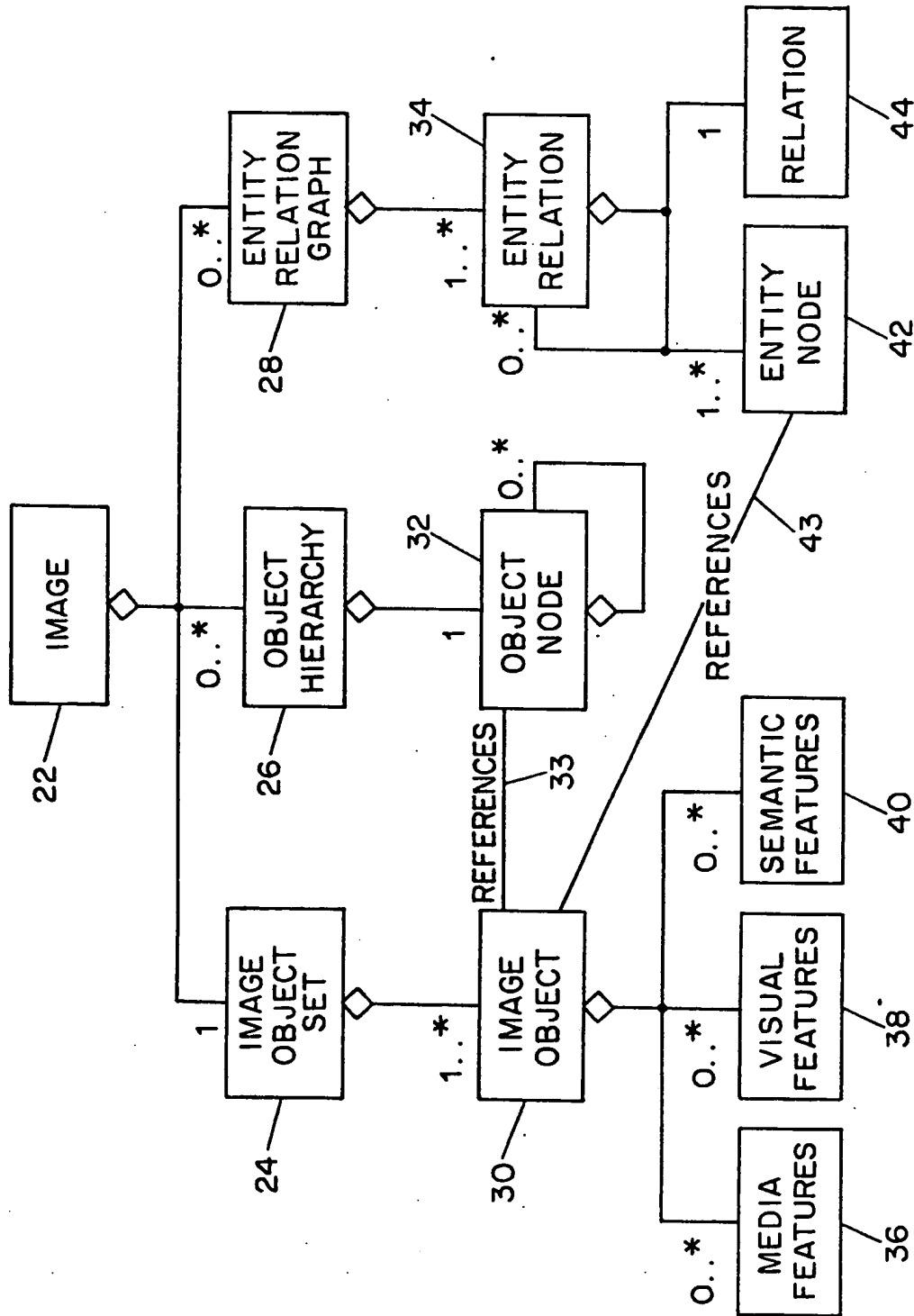


FIG. 2

3/12

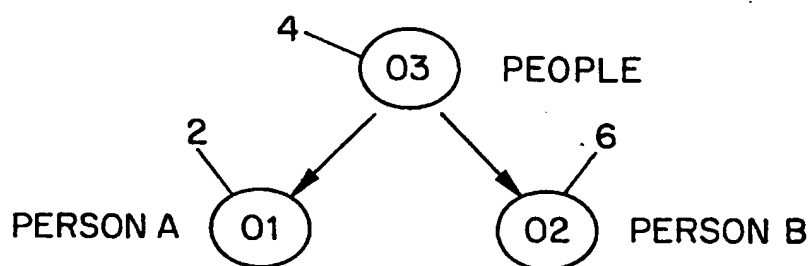


FIG. 3a

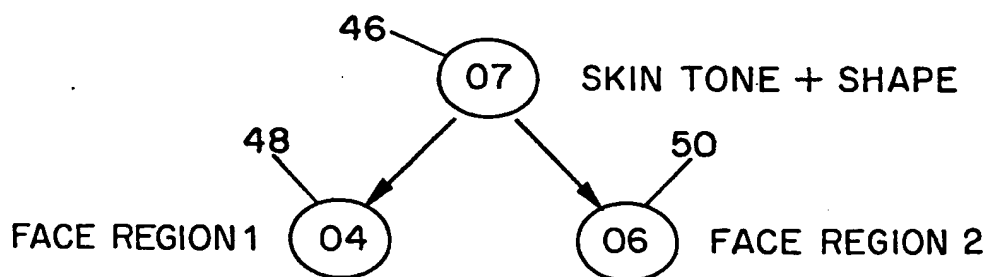


FIG. 3b

4/12

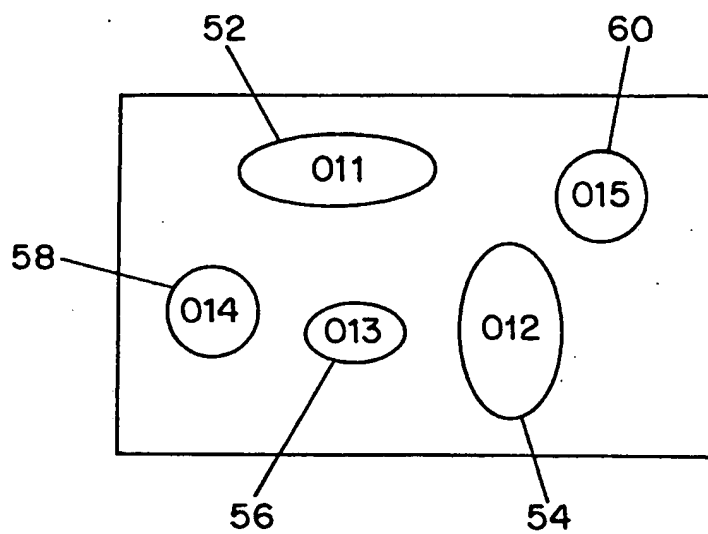


FIG. 4a

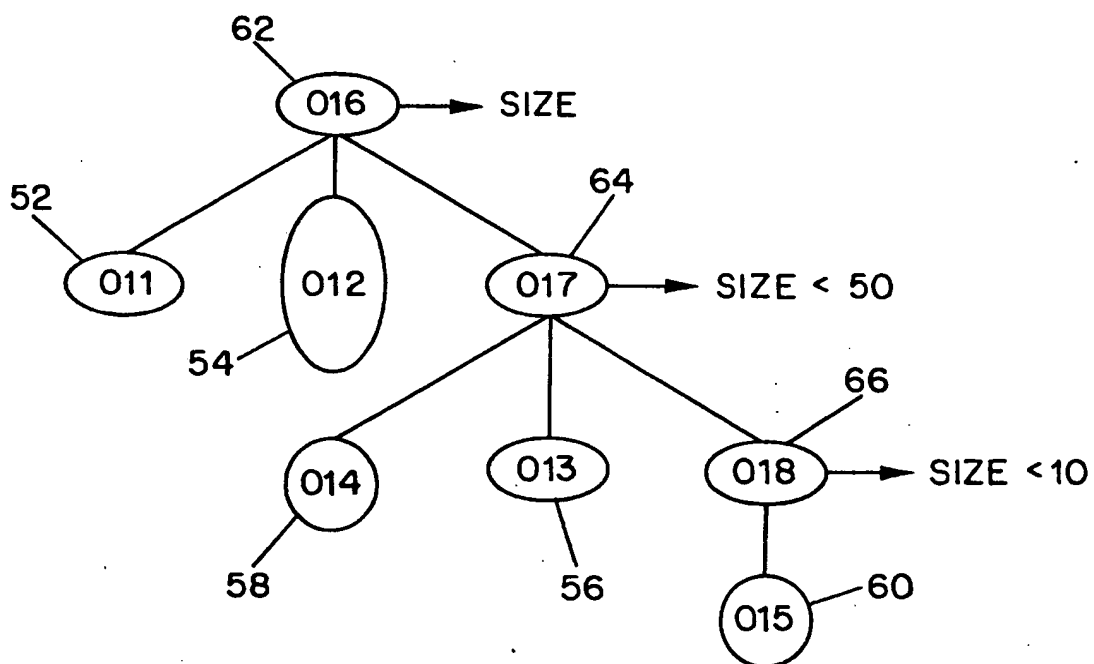


FIG. 4b

5/12

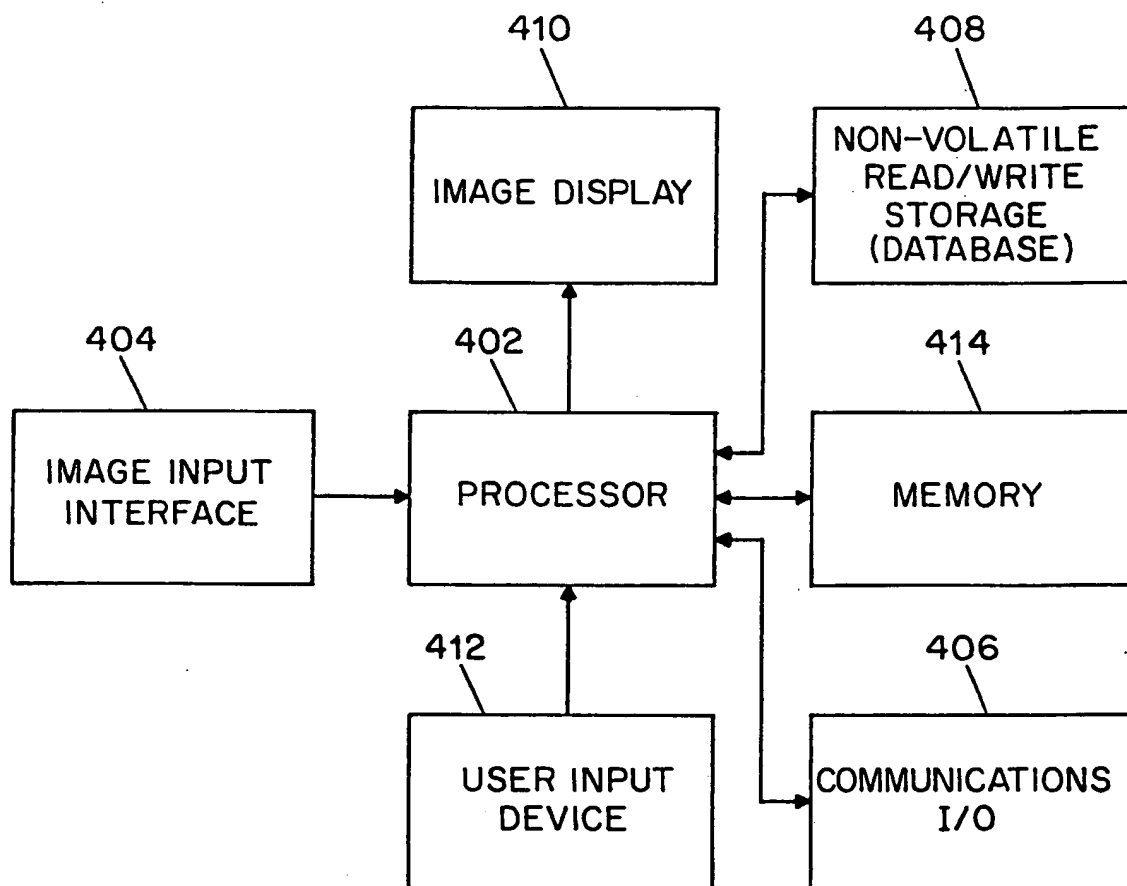


FIG. 5

6/12

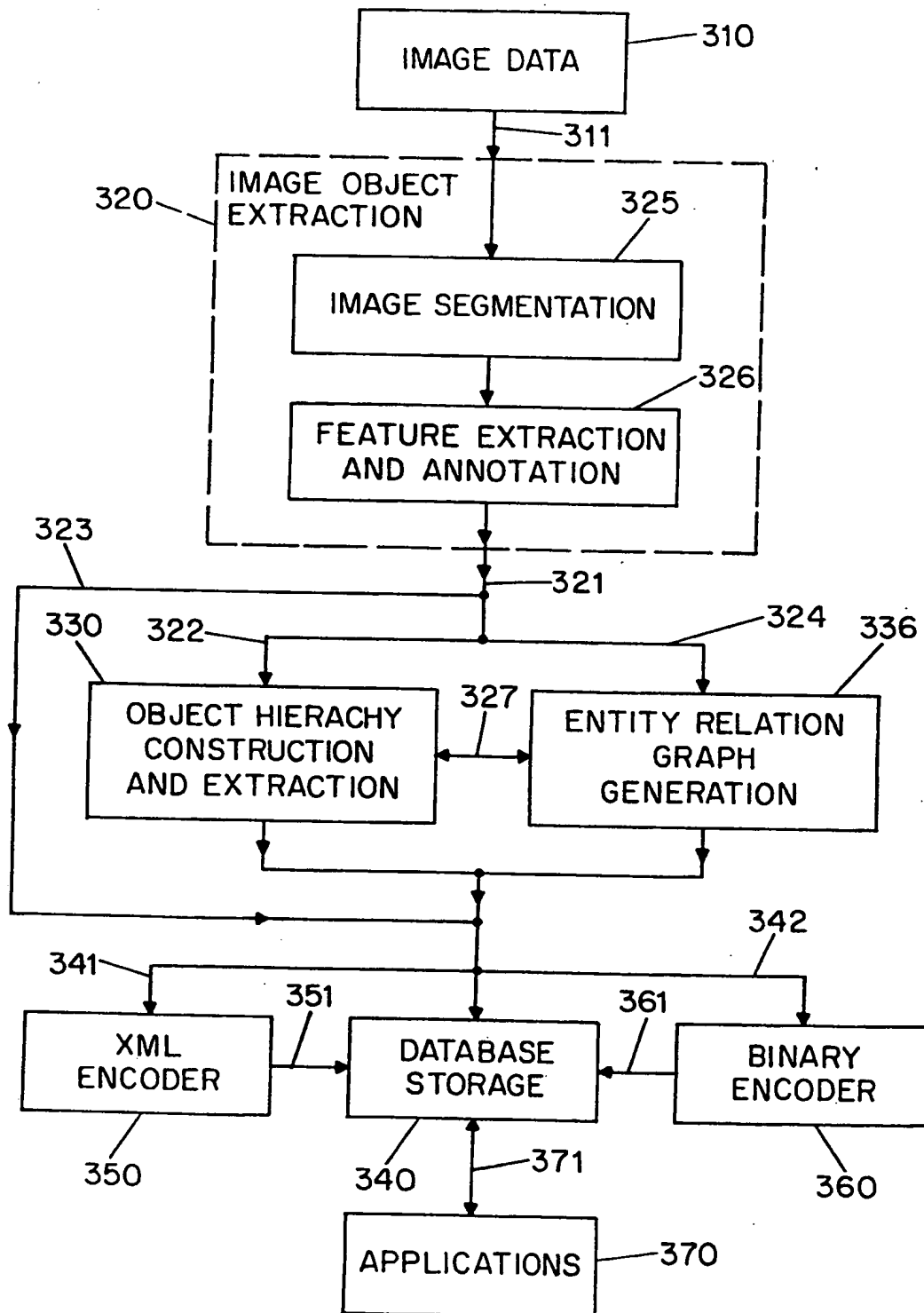


FIG. 6

7/12

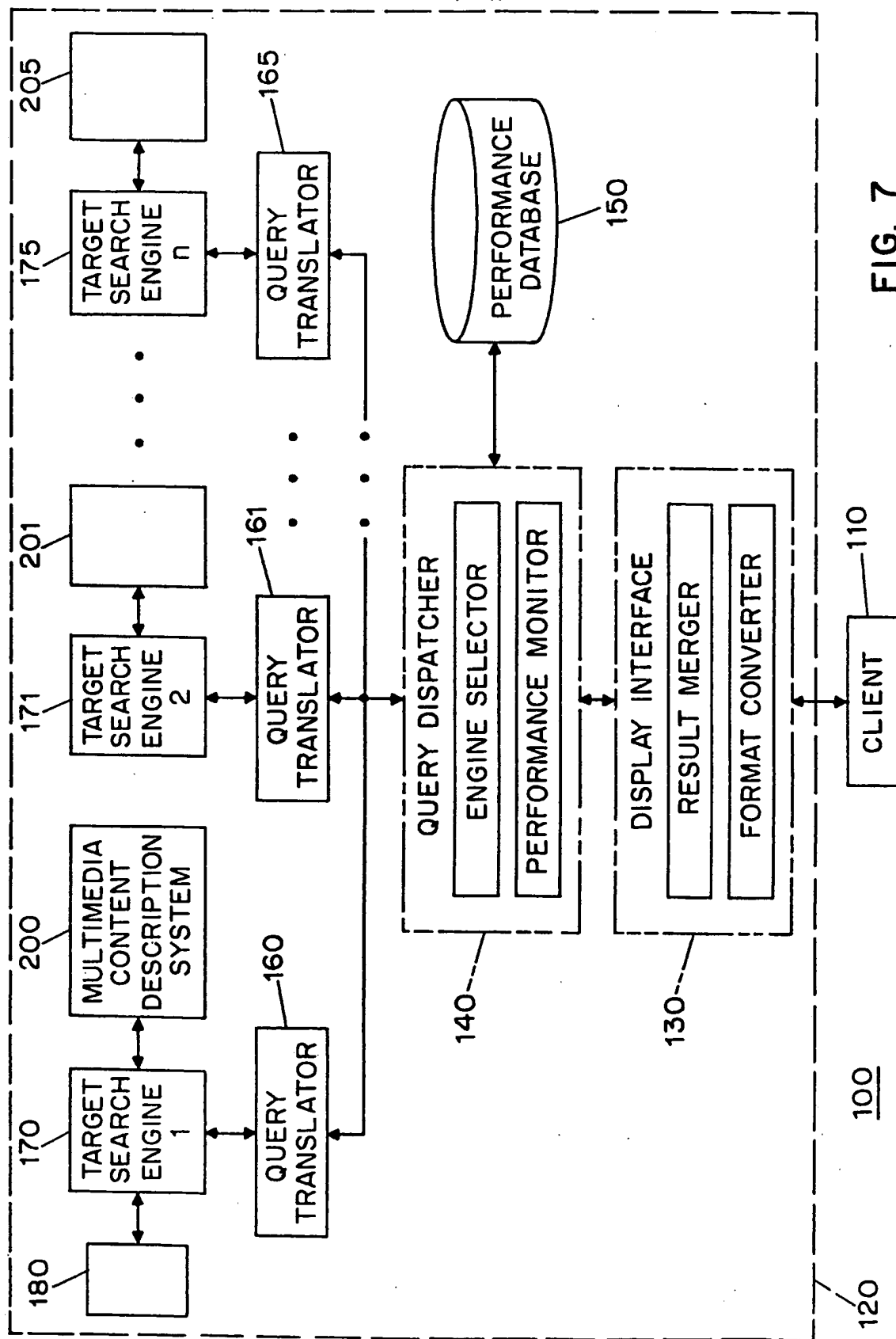


FIG. 7

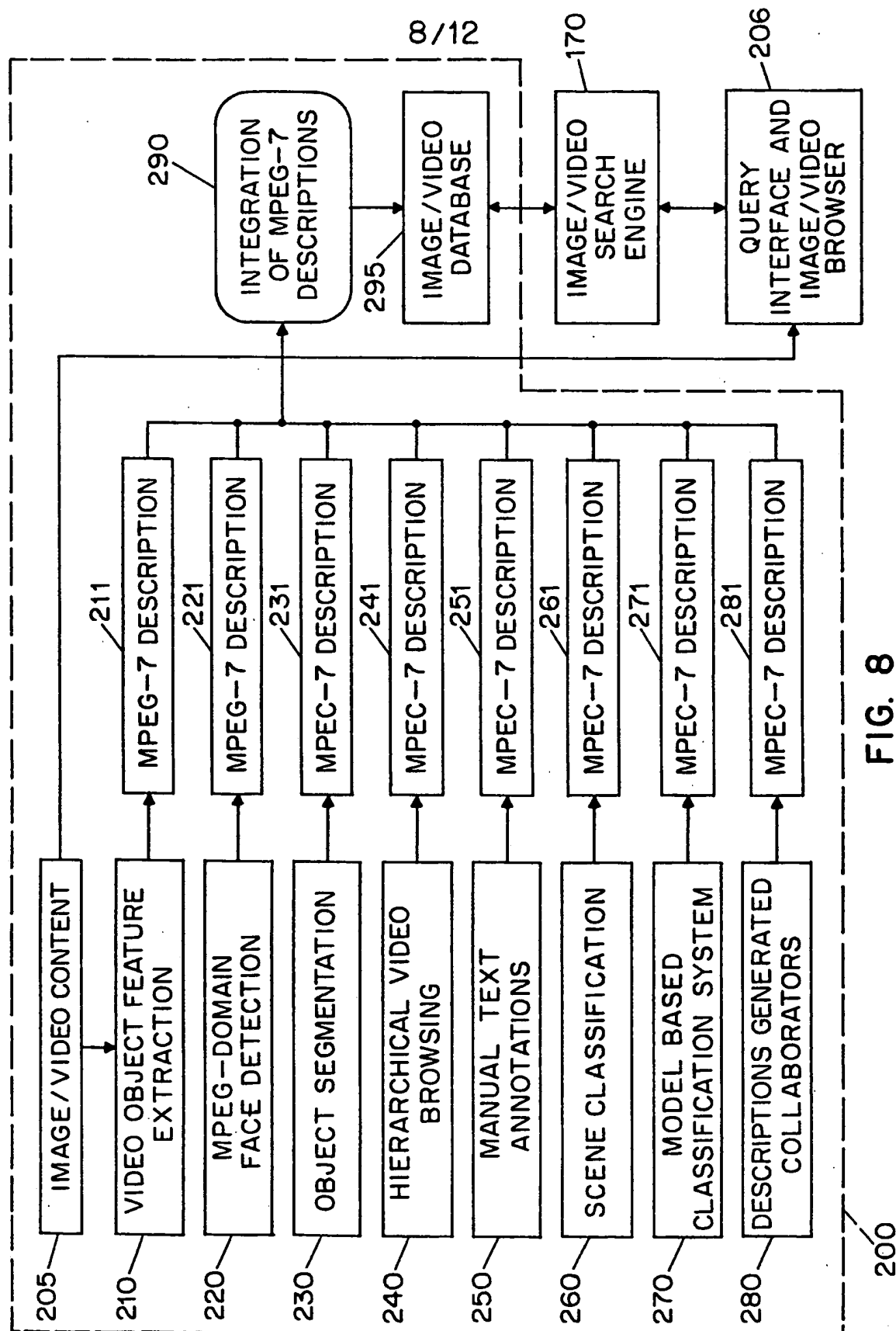


FIG. 8

9/12

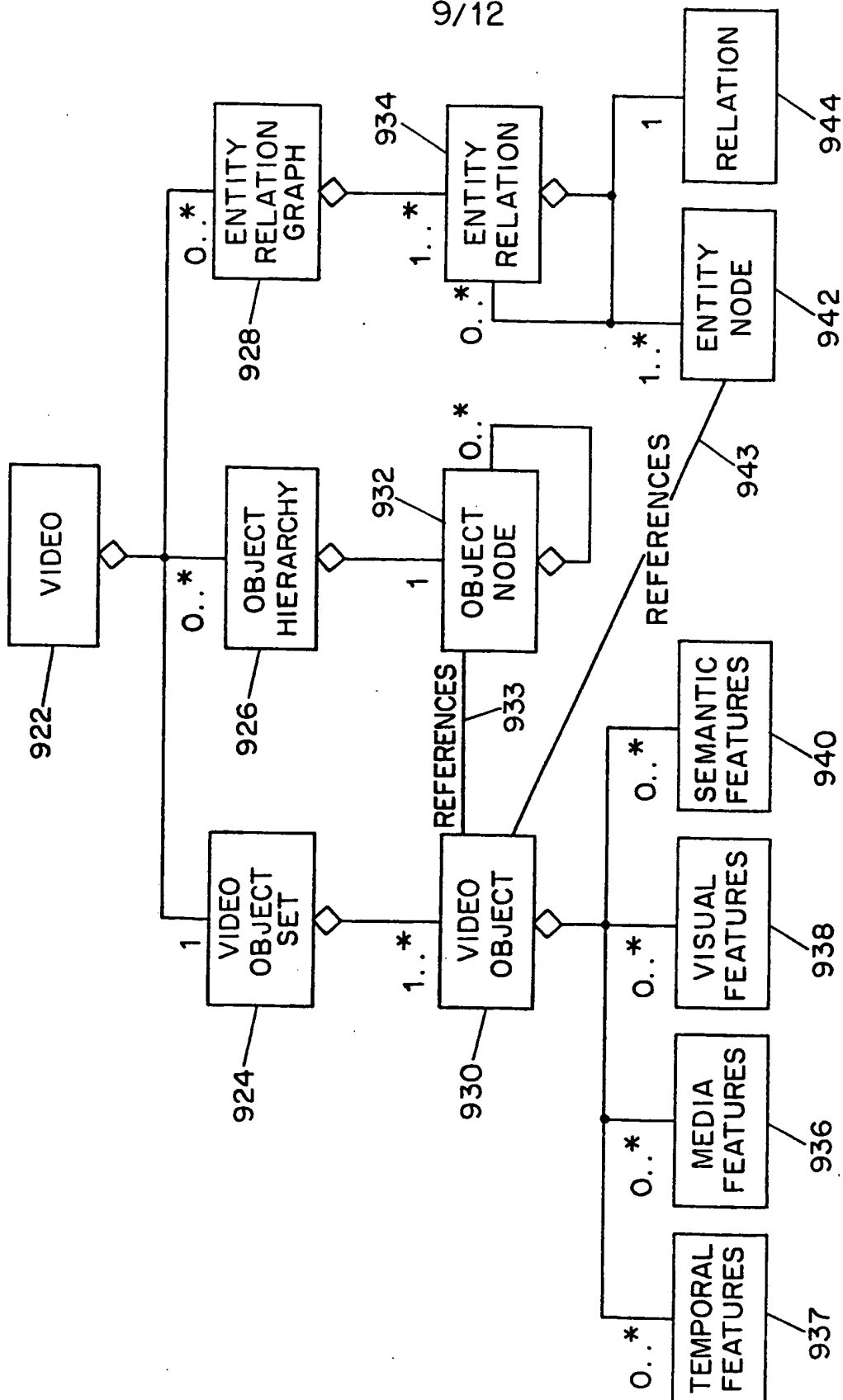


FIG. 9

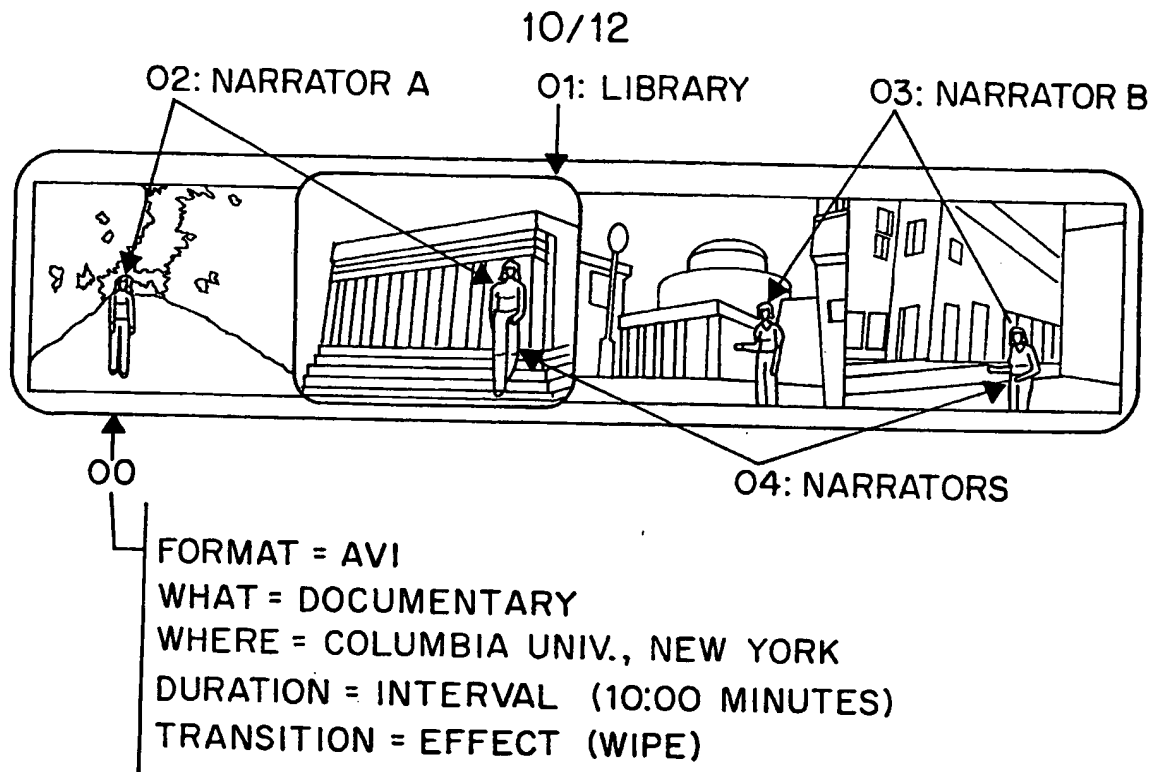


FIG. 10

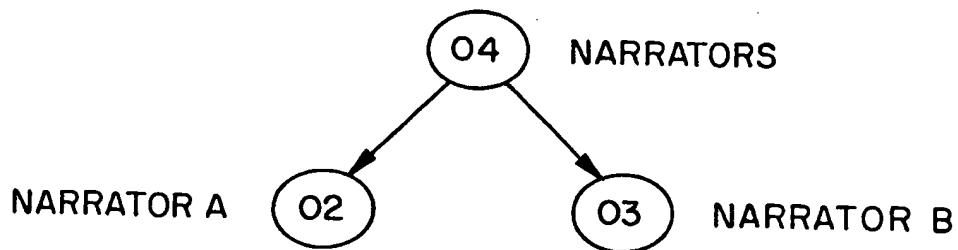


FIG. 11

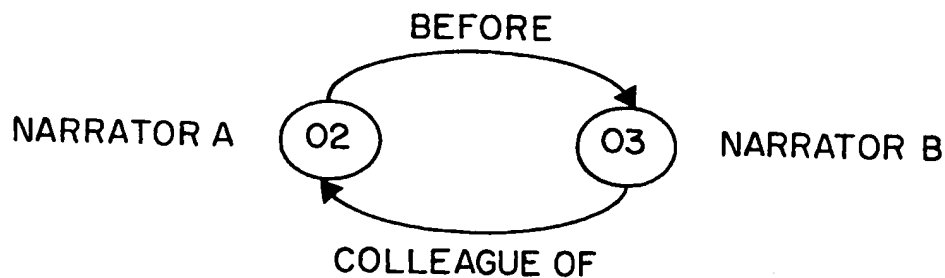


FIG. 12

11/12

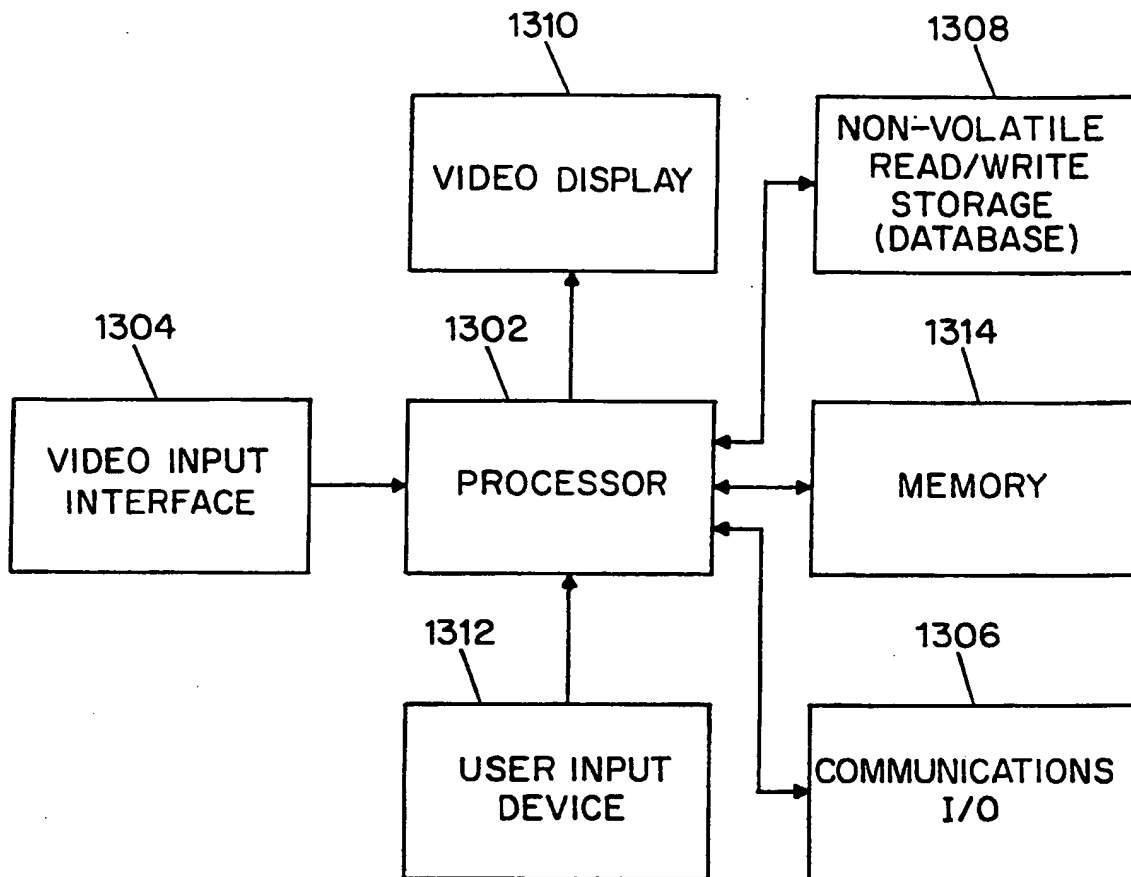
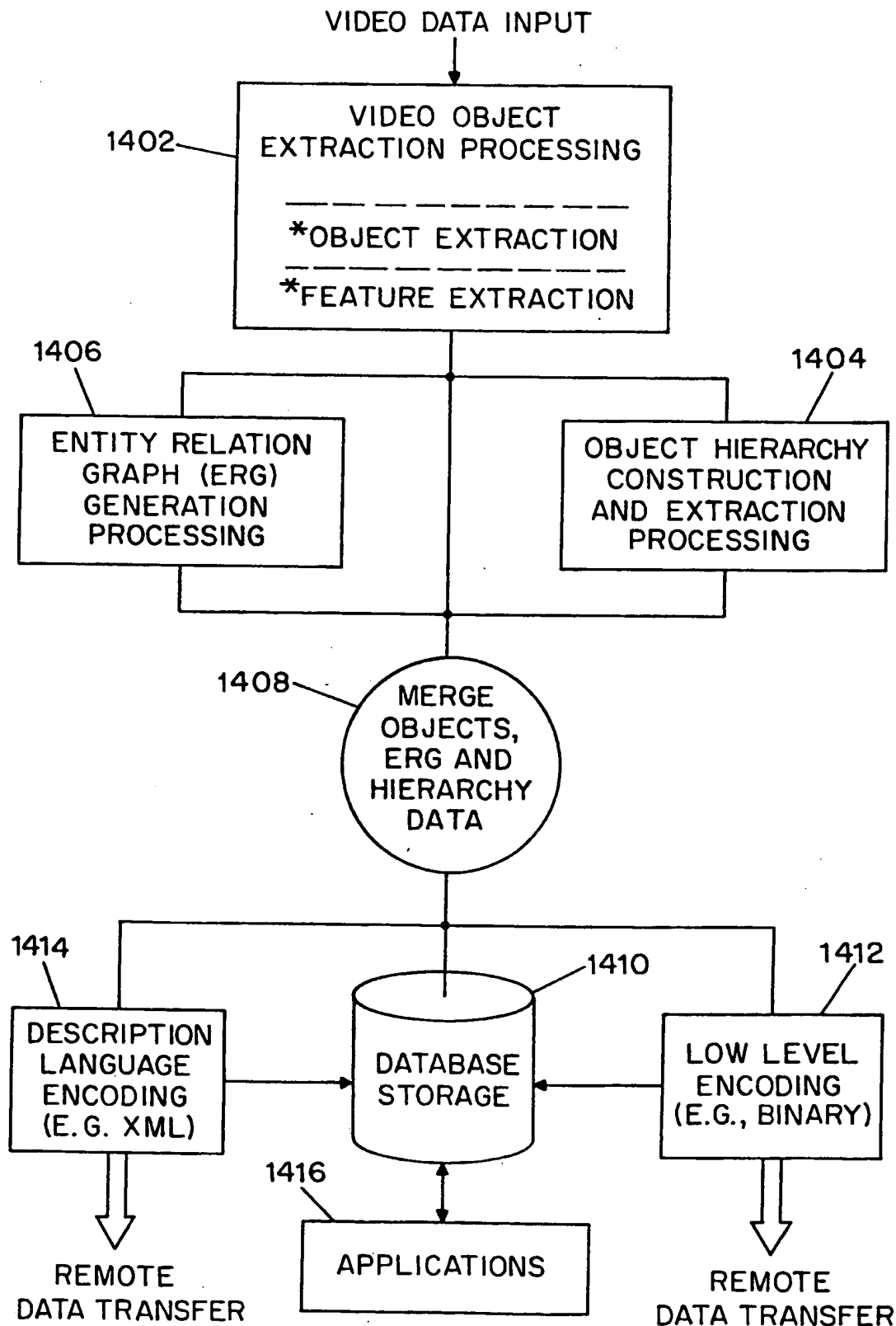


FIG. 13

**FIG. 14**

APPENDIX A: Document Type Definition of Video Description Scheme**video_ds.dtd:**

```

<!-- Video DS -->

<!-- Entities are like macros. They can be referenced by using the notation "%EntityName;" -->
>
<!-- For clarity, we have chosen not to reference them in this DTD, although some of them
are derived from the Image DTD -->
<!-- Some of the elements in this DTD are inherited from the image DTD: object_hierarchy,
entity_relation_graph, etc. We have included some of them in this annex. -->
<!ENTITY % video_object_elements "
    vid_obj_media_features?,
    vid_obj_semantic_features?,
    vid_obj_visual_features?,
    vid_obj_temporal_features?">

<!ENTITY % ref_video_object_attributes "
    %image_object_attributes;">

<!ENTITY % only_vid_obj_media_features_elements "
    bit_rate?">

<!ENTITY % vid_obj_media_features_elements "
    %img_obj_media_features_elements;
    %only_vid_obj_media_features_elements;">

<!ENTITY % vid_obj_semantic_features_elements "
    %img_obj_semantic_features_elements;">

<!ENTITY % only_vid_obj_visual_features_elements "
    video_scl?, visual_sprite?, transition?, camera_motion?, size?, key_frame*">

<!ENTITY % vid_obj_visual_features_elements "
    %img_obj_visual_features_elements;
    %only_vid_obj_visual_features_elements;">

<!ENTITY % vid_obj_temporal_features_elements "
    time?">

```

```

<!ELEMENT video (video_object_set, event_hierarchy*, entity_relation_graph*)>
<!ATTLIST video
    id ID #IMPLIED>

<!ELEMENT video_object_set (video_object+)>

<!ELEMENT video_object (vid_obj_media_features?, vid_obj_semantic_features?,
    vid_obj_visual_features?, vid_obj_temporal_features?)>
<!ATTLIST video_object
    type (LOCAL|SEGMENT|GLOBAL) #REQUIRED
    id ID #IMPLIED
    object_ref IDREF #IMPLIED
    object_node_ref IDREFS #IMPLIED
    entity_node_ref IDREFS #IMPLIED>

<!ELEMENT vid_obj_media_features (
    bit_rate?,
    location?, file_format?, file_size?, resolution?, modality_transcoding?)>

<!ELEMENT vid_obj_semantic_features (text_annotation?, who?, what_action?, where?,
    why?,
    when?)>

<!ELEMENT vid_obj_visual_features (
    image_scl?, color?, texture?, shape?, size?, position?, motion?,
    video_scl?, visual_sprite?, transition?, camera_motion?, size?, key_frame*)>

<!ELEMENT vid_obj_temporal_features (time?)>

<!-- The object hierarchy and the entity relation graph are defined in the Image DS (Proposal
# 480). We include them in this DTD for convenience. -->

<!-- Object hierarchy element -->
<!-- The attribute type is the hierarchy binding type -->
<!ELEMENT object_hierarchy (object_node)>
<!ATTLIST object_hierarchy
    id ID #IMPLIED
    type CDATA #IMPLIED>

```

```
<!ELEMENT object_node (object_node*)>
<!ATTLIST object_node
    id ID #IMPLIED
    object_ref IDREF #REQUIRED>

<!-- Entity relation graph element-->
<!-- Possible types of entity relations and entity relation graphs:
    - Spatial: topological, directional
    - Temporal: topological, directional
    - Semantic -->
<!ELEMENT entity_relation_graph (entity_relation+)>
<!ATTLIST entity_relation_graph
    id ID #IMPLIED
    type CDATA #IMPLIED>

<!ELEMENT entity_relation (relation, (entity_node | entity_node_set | entity_relation)*)>
<!ATTLIST entity_relation
    type CDATA #IMPLIED>

<!ELEMENT relation (#PCDATA | code)*>

<!ELEMENT entity_node (#PCDATA)>
<!ATTLIST entity_node
    id ID #IMPLIED
    object_ref IDREF #REQUIRED>

<!ELEMENT entity_node_set (entity_node+)>

<!-- External image DS DTD -->
<!ENTITY % image_ds SYSTEM "image_ds.dtd">
%image_ds;

<!-- External scalable video DTD -->
<!ENTITY % video_scl SYSTEM "video_scl.dtd">
%video_scl;

<!-- External visual sprite DTD -->
<!ENTITY % visual_sprite SYSTEM "visual_sprite.dtd">
%visual_sprite;
```

```

<!-- External transition DTD -->
<!ENTITY % transition SYSTEM "transition.dtd">
%transition;

<!-- External camera motion DTD -->
<!ENTITY % camera_motion SYSTEM "camera_motion.dtd">
%camera_motion;

<!ELEMENT key_frame (size_dimensions?, time_instant?)>

<!-- Video DS end -->

```

location.dtd:

```

<!-- Description of resources' location-->

<!-- Objects, image, videos can be located/accessed at different locations -->
<!ELEMENT location (location_site*)>

<!-- One location site -->
<!ELEMENT location_site EMPTY>
<!ATTLIST location_site
    href CDATA #REQUIRED
    title CDATA #IMPLIED>

<!ELEMENT code (location*)>
<!ATTLIST code
    type (EXTRACTION|DISTANCE) "EXTRACTION"
    language (C|JAVA|PERL) #REQUIRED
    version CDATA #REQUIRED>

<!-- Description of resources' storage location -->

```

video_scl.dtd:

```

<!-- Video scalability features -->

<!ELEMENT video_scl(video_sclobj, code*)
<!ELEMENT video_sclobj(vid_obj_scltype, vid_obj_mode, vid_obj_numlayers, coderef,
subsamp_factor, vid_obj_shape?)>

```

```

<!ELEMENT vid_obj_scltype EMPTY>
<!ATTLIST vid_obj_scltype
    typeinfo (DATPARTITION|SPATIAL|TEMPORAL|SNR) #REQUIRED>

<!-- Video scalability (subtype) mode features -->
<!ELEMENT vid_obj_mode EMPTY>
<!ATTLIST vid_obj_mode
    modeinfo CDATA #REQUIRED>

<!ELEMENT numlayers EMPTY>
<!ATTLIST vid_obj_numlayers
    numval #REQUIRED>

<!ELEMENT coderef EMPTY>
<!ATTLIST coderef
    layernum #REQUIRED>

<-- subsampling ratio n/m for horizontal and vertical directions -->
<!ELEMENT subsamp_factor EMPTY>
<!ATTLIST subsamp_factor
    hor_factor_n CDATA #REQUIRED
    hor_factor_m CDATA #REQUIRED
    vert_factor_n CDATA #REQUIRED
    vert_factor_m CDATA #REQUIRED>

<!ELEMENT vid_obj_shape(shape)>

<!-- scalability features end -->

```

visual_sprite.dtd:

```

<!-- visual sprite features -->

<!ELEMENT visual_sprite (vis_spriteobj)>

<!ELEMENT vis_spriteobj (vis_spriteobj_info, code*)>

<!ELEMENT vis_spriteobj_info (vis_sprite_dim, vis_sprite_shape, vis_sprite_trajectory,
vis_sprite_warp, vis_sprite_brightness, vis_sprite_texture)>

<!ELEMENT vis_sprite_dim (sprite_size, sprite_num_pts, sprite_coord)>

```

```

<!ELEMENT sprite_size EMPTY>
<!-- ATTLIST sprite_size
      sprite_width CDATA #REQUIRED
      sprite_height CDATA #REQUIRED -->

<!ELEMENT sprite_num_pts EMPTY>
<!-- ATTLIST sprite_num_pts
      num_pts CDATA #REQUIRED -->

<!ELEMENT sprite_coord EMPTY>
<!-- ATTLIST sprite_coord
      xcoord CDATA #REQUIRED
      ycoord CDATA #REQUIRED -->

<!ELEMENT vis_sprite_shape (shape)>

<!ELEMENT vis_sprite_trajectory (motion)>

<!ELEMENT vis_sprite_warp EMPTY>
<!-- ATTLIST vis_sprite_warp
      num_pts CDATA #REQUIRED -->

<!ELEMENT vis_sprite_brightness EMPTY>
<!-- ATTLIST vis_sprite_brightness
      avgbright CDATA #REQUIRED
      varbright CDATA #REQUIRED -->

<!ELEMENT vis_sprite_texture (texture)>

```

transition.dtd:

```

<!-- Transition features -->

<!ELEMENT transition (transition_desc1*)>

<!ELEMENT transition_desc1 (transition_desc1_value, code*)>
<!ELEMENT transition_desc1_value (effect)>
<!ELEMENT effect (#PCDATA)>

<!-- Transition features -->

```

camera_motion.dtd:

```

<!-- Camera motion features -->

<!ELEMENT camera_motion (background_affine_motion*)>

<!-- Affine model for camera motion detection -->
<!ELEMENT background_affine_motion (background_affine_motion_value, code*)>

<!ELEMENT background_affine_motion_value (panning?, zoom?)>

<!ELEMENT panning (direction?)>
<!ATTLIST panning
    direction (NT|NE|ET|SE|ST|SW|WT|NW) #REQUIRED>

<!ELEMENT zoom (direction?)>
<!ATTLIST zoom
    direction (IN|OUT) #REQUIRED>

<!ELEMENT direction (#PCDATA)>

<!-- Camera motion features end -->

```

motion.dtd:

```

<!-- Motion features -->

<!ELEMENT motion (affine_model*)>

<!-- Affine motion feature -->
<!ELEMENT affine_model (affine_model_value*, code*)>

<!ELEMENT affine_model_value (parameters?, trajectory?)>

<!ELEMENT parameters (affine_bin*)>
<!ATTLIST parameters
    length CDATA #IMPLIED>

<!ELEMENT affine_bin (#PCDATA)>
<!ATTLIST affine_bin

```

```

    value CDATA #REQUIRED>

<!ELEMENT trajectory (vector2d*)>
<!ATTLIST trajectory
    length CDATA #IMPLIED>

<!ELEMENT vector2d (vect_x, vect_y)?>
<!ATTLIST vector2d
    x CDATA #REQUIRED
    y CDATA #REQUIRED>

<!ELEMENT vect_x (#PCDATA)>
<!ELEMENT vect_y (#PCDATA)>

<!-- Motion features end -->

```

time.dtd:

```

<!-- Time duration features -->

<!ELEMENT time (time_interval* | time_instant | time_span)>

<!-- Continuous duration time (seconds or frames in video clip)-->
<!ELEMENT time_interval (start, end)?>
<!ATTLIST time_interval
    unit (SECONDS|FRAMES) "SECONDS"
    start CDATA #REQUIRED
    end CDATA #REQUIRED>

<!ELEMENT start (#PCDATA)>
<!ELEMENT end (#PCDATA)>

<!-- Instant in time -->
<!ELEMENT time_instant (instant)?>
<!ATTLIST time_instant
    unit (SECONDS|FRAMES) "SECONDS"
    instant CDATA #REQUIRED>

<!ELEMENT instant (#PCDATA)>

```

```
<!-- Continuous duration time -->  
<!ELEMENT time_span (span)?>  
<!ATTLIST time_span  
    unit (SECONDS|FRAMES) "SECONDS"  
    span CDATA #REQUIRED>  
  
<!ELEMENT span (#PCDATA)>  
  
<!-- Time duration features end -->
```